

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DESARROLLO DE UN PROTOTIPO DE SOFTWARE DISTRIBUIDO PARA LA GESTIÓN DE PRODUCTOS Y SERVICIOS OFERTADOS POR LA ASOCIACIÓN DE ESTUDIANTES DE INGENIERÍA ELECTRÓNICA

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

RICARDO DAVID TAIPE HIDALGO

ricardo.taipe@epn.edu.ec

DIRECTOR: PhD. ANA MARÍA ZAMBRANO VIZUETE

ana.zambrano@epn.edu.ec

Quito, diciembre 2019

AVAL

Certifico que el presente trabajo fue desarrollado por Ricardo David Taipe Hidalgo, bajo mi supervisión.

PhD. ANA ZAMBRANO
DIRECTORA DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo, Ricardo David Taipe Hidalgo, 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.

RICARDO DAVID TAIPE HIDALGO

DEDICATORIA

Dedico este Trabajo de Titulación a mis padres y hermanos, mi familia, quienes han sido la base de mi formación, cada uno de ustedes ha aportado grandes cosas y lecciones a mi vida, y me han ayudado a enfrentar el gran reto de enfrentarme a la sociedad. Les agradezco por todo, en especial por ser los principales benefactores del desarrollo y culminación de mi tesis.

Ricardo Taípe

AGRADECIMIENTO

A mis padres quienes son mi motor y mi mayor inspiración, que a través de su amor, paciencia y buenos valores, ayudan a trazar mi camino.

A mis Hermanos que con sus palabras me hacían sentir orgulloso de quién que soy. Ojala algún día yo me convierta en esa fuerza para que puedan seguir avanzando en su camino.

A mis amigos. Con todos los que compartí dentro y fuera de las aulas. Aquellos amigos que hicieron que la vida universitaria se convierta en una gran experiencia y pueda llevar conmigo excelentes recuerdos.

Finalmente quiero expresar un gran agradecimiento a PhD. Ana María Zambrano, principal colaboradora durante todo este proceso, quien con su dirección, conocimiento y colaboración permitió el desarrollo de este Trabajo de Titulación.

ÍNDICE DE 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.....	1
1.1 OBJETIVOS	1
1.2 ALCANCE	1
1.3 MARCO TEÓRICO.....	4
1.3.1 APLICACIONES MÓVILES Y SMARTPHONES	4
1.3.2 CLOUD COMPUTING	5
1.3.3 DESCRIPCIÓN DE HERRAMIENTAS Y TECNOLOGÍAS POR USAR EN LA CAPA DE DATOS	11
1.3.4 DESCRIPCIÓN DE HERRAMIENTAS Y TECNOLOGÍAS POR USAR EN LA CAPA DE LÓGICA DE NEGOCIO.....	12
1.3.5 DESCRIPCIÓN DE HERRAMIENTAS Y TECNOLOGÍAS POR USAR EN LA CAPA DE PRESENTACIÓN.....	15
1.3.6 METODOLOGÍA ÁGIL DE DESARROLLO KANBAN	17
2 METODOLOGÍA.....	18
2.1 SITUACIÓN ACTUAL DE LOS PRODUCTOS Y SERVICIOS OFERTADOS POR LA ASOCIACIÓN DE ESTUDIANTES DE INGENIERÍA ELECTRÓNICA (AEIE).....	18
2.2 DISEÑO	19
2.2.1 TABLERO DE ACTIVIDADES KANBAN.....	19
2.2.2 ANÁLISIS DE REQUERIMIENTOS	20
2.2.3 SELECCIÓN DEL PROVEEDOR CLOUD COMPUTING.....	24
2.2.4 DISEÑO DE LA CAPA DE DATOS.....	26
2.2.5 DISEÑO DE LA CAPA DE LÓGICA DE NEGOCIOS.....	28

2.2.6	DISEÑO DE LA CAPA DE PRESENTACIÓN	36
2.3	IMPLEMENTACIÓN	42
2.3.1	ACTUALIZACIÓN DEL TABLERO KANBAN	42
2.3.2	INSTALACIÓN DE LAS HERRAMIENTAS NECESARIAS	43
2.3.3	IMPLEMENTACIÓN DE LA CAPA DE DATOS.....	48
2.3.4	IMPLEMENTACIÓN DE LA CAPA DE LÓGICA DE NEGOCIO	49
2.3.5	IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN	52
3	RESULTADOS Y DISCUSIÓN	60
3.1	ACTUALIZACIÓN TABLERO KANBAN	60
3.2	PRUEBAS DE VALIDACIÓN DE REQUERIMIENTOS FUNCIONALES.	61
3.2.1	PRUEBA DE VALIDACIÓN DEL MÓDULO CATÁLOGO.....	61
3.2.2	PRUEBA DE VALIDACIÓN DEL MÓDULO VENTAS	66
3.2.3	PRUEBA DE VALIDACIÓN DEL MÓDULO SERVICIOS	68
3.2.4	PRUEBA DE VALIDACIÓN DEL MÓDULO CURSOS	75
3.2.5	PRUEBA DE VALIDACIÓN DEL MÓDULO INFORMATIVO.....	77
3.2.6	PRUEBA DE VALIDACIÓN DEL MÓDULO USUARIOS.....	79
3.2.7	PRUEBAS DE VALIDACIÓN DEL MÓDULO MÓVIL	82
3.3	PRUEBAS DE VALIDACIÓN DE REQUERIMIENTOS NO FUNCIONALES.....	86
3.3.1	RNF01: USABILIDAD	86
3.3.2	RN02: ESCALABILIDAD	87
3.3.3	RNF03: DISPONIBILIDAD.....	88
3.3.4	RNF04: AUTENTICACIÓN	88
3.4	TABLERO KANBAN FINAL.....	89
4	CONCLUSIONES Y RECOMENDACIONES	91
4.1.	CONCLUSIONES.....	91
4.2.	RECOMENDACIONES	92
5	REFERENCIAS BIBLIOGRÁFICAS	94
	ANEXOS	98

ÍNDICE DE FIGURAS

Figura 1.1 Esquema del prototipo de sistema distribuido para la gestión de productos y servicios ofertados por la AEIE.	2
Figura 1.2 Ventas a nivel mundial de teléfonos inteligentes.	5
Figura 1.3 Software como Servicio.	7
Figura 1.4 Plataforma como Servicio.	8
Figura 1.5 Infraestructura como Servicio.	8
Figura 1.6 Tablero Kanban para Desarrollo de Software.	17
Figura 2.1 Tablero Kanban para la fase de Diseño.	20
Figura 2.2 Diagrama de colecciones para la base de datos.	27
Figura 2.3 Actores que interactúan con el sistema.	29
Figura 2.4 Caso de uso del módulo Catálogo.	30
Figura 2.5 Caso de uso del módulo Ventas.	31
Figura 2.6 Caso de uso del módulo Servicios.	32
Figura 2.7 Caso de uso del módulo Cursos.	32
Figura 2.8 Caso de uso del módulo Informativo.	33
Figura 2.9 Caso de uso del módulo Usuarios.	33
Figura 2.10 Caso de uso del módulo Móvil.	34
Figura 2.11 Diagrama de clases.	35
Figura 2.12 Interfaz para el inicio de sesión.	37
Figura 2.13 Interfaz para el módulo Catálogo.	37
Figura 2.14 Interfaz para el módulo Ventas.	38
Figura 2.15 Interfaz para el módulo Servicios.	38
Figura 2.16 Interfaz para el módulo Cursos.	39
Figura 2.17 Interfaz para el módulo Informativo.	39
Figura 2.18 Interfaz para el módulo Usuarios.	40
Figura 2.19 Interfaz módulo Móvil de los menús “ <i>Productos</i> ” y “ <i>Casilleros</i> ”.	41
Figura 2.20 Interfaz módulo Móvil de los menús “ <i>Cursos</i> ” y “ <i>Anuncios</i> ”.	41
Figura 2.21 Tablero Kanban para la fase de implementación.	42
Figura 2.22 Creación de un recurso a través del Portal Microsoft Azure.	43
Figura 2.23 Creación de una máquina virtual en Microsoft Azure.	43
Figura 2.24 Conceptos básicos de una máquina virtual en Azure.	44
Figura 2.25 Detalles de instancia de una máquina virtual en Azure.	44
Figura 2.26 Configuración del acceso remoto a la máquina virtual Azure.	45
Figura 2.27 Reglas de puerto de entrada para la máquina virtual.	45
Figura 2.28 Prueba de conexión remota mediante SSH.	45

Figura 2.29	Pantalla de bienvenida en Visual Studio Code.	47
Figura 2.30	Pantalla de Bienvenida de Android Studio.....	47
Figura 2.31	Carpeta del servicio web.	50
Figura 2.32	Implementación de la interfaz gráfica del Módulo Catálogo.....	53
Figura 2.33	Menú del Módulo Catálogo.....	53
Figura 2.34	Interfaz gráfica del " <i>ProductComponent</i> ".	55
Figura 2.35	Interfaces de la aplicación móvil.....	56
Figura 2.36	Interfaz módulo Móvil correspondiente a los menú Cursos y Anuncios.	58
Figura 2.37	Interfaz módulo Móvil correspondiente a los menú Cursos y Anuncios.	59
Figura 3.1	Tablero Kanban para la fase de pruebas.....	60
Figura 3.2	Creación de una categoría.	61
Figura 3.3	Acciones Editar y Eliminar.....	62
Figura 3.4	Mensaje de actualización exitosa para una categoría.	62
Figura 3.5	Mensaje de eliminación exitosa para una categoría.	62
Figura 3.6	Creación de un proveedor.	63
Figura 3.7	Acciones de eliminar y actualizar.	63
Figura 3.8	Mensaje de actualización exitosa para un proveedor.	63
Figura 3.9	Mensaje de eliminación exitosa para un proveedor.....	64
Figura 3.10	Creación de un producto.	64
Figura 3.11	Acciones de eliminar y actualizar.	65
Figura 3.12	Mensaje de actualización exitosa para un producto.	65
Figura 3.13	Mensaje de eliminación exitosa de un producto.	65
Figura 3.14	Selección de un producto para la venta.....	66
Figura 3.15	Mensaje de información al usuario.	66
Figura 3.16	Venta de productos.	67
Figura 3.17	Mensaje de registro de venta exitoso.	67
Figura 3.18	Lista de ventas registradas.....	67
Figura 3.19	Registro de un semestre referencial.	68
Figura 3.20	Creación de una categoría.	69
Figura 3.21	Acciones Editar y Eliminar.....	69
Figura 3.22	Mensaje de actualización exitosa.....	70
Figura 3.23	Mensaje de eliminación exitosa para una categoría.	70
Figura 3.24	Registro de aporte.....	71
Figura 3.25	Mensaje de registro exitoso del aporte.	71
Figura 3.26	Eliminación de un aporte.	71
Figura 3.27	Creación de un casillero.....	72

Figura 3.28 Acciones Editar y Eliminar.....	72
Figura 3.29 Mensaje de actualización exitosa para una categoría.	72
Figura 3.30 Mensaje de eliminación exitosa de un casillero.	73
Figura 3.31 Alquiler de un casillero.	73
Figura 3.32 Mensaje de alquiler exitoso.	73
Figura 3.33 Acciones disponibles sobre un casillero alquilado.	74
Figura 3.34 Mensaje de notificación enviada con éxito.	74
Figura 3.35 Correo electrónico que recibe el estudiante.	74
Figura 3.36 Eliminación de alquiler.	75
Figura 3.37 Creación de un curso.	75
Figura 3.38 Acciones Editar y Eliminar.....	76
Figura 3.39 Mensaje de actualización exitosa para un curso.	76
Figura 3.40 Mensaje de eliminación exitosa para una categoría.	76
Figura 3.41 Registro de un estudiante en un curso.	77
Figura 3.42 Publicación de un anuncio de tipo informativo.....	78
Figura 3.43 Acciones Editar y Eliminar.....	78
Figura 3.44 Mensaje de actualización exitosa para un anuncio.....	78
Figura 3.45 Mensaje de eliminación exitosa para un anuncio.	79
Figura 3.46 Creación de un nuevo usuario.....	80
Figura 3.47 Administrador con permisos y Usuario sin permiso de uso del módulo "Usuarios".....	80
Figura 3.48 Acciones Editar y Eliminar.....	80
Figura 3.49 Mensaje de actualización exitosa de un usuario.....	81
Figura 3.50 Mensaje de eliminación exitosa para un usuario.	81
Figura 3.51 Lista de usuarios.	82
Figura 3.52 Interfaz módulo Móvil correspondiente a los menú Productos y Casilleros...82	82
Figura 3.53 Interfaz módulo Móvil correspondiente a los menú Cursos.....83	83
Figura 3.54 Interfaz módulo Móvil correspondiente a los menú Anuncios.83	83
Figura 3.55 Respuesta de la pregunta 1 en la encuesta de satisfacción de la aplicación móvil.	87
Figura 3.56 Respuesta de la pregunta 1 en la encuesta de satisfacción de la aplicación web.....	87
Figura 3.57 Interfaz de inicio de sesión.	88
Figura 3.58 Nombre de usuario en la página principal.	89
Figura 3.59 Mensaje de autenticación fallida.	89
Figura 3.60 Tablero Final Kanban.	90

ÍNDICE DE TABLAS

Tabla 1.1	Ventas mundiales de <i>smartphones</i> a usuarios finales por sistema operativo[4].	5
Tabla 1.2	Características esenciales del <i>Cloud Computing</i> .	6
Tabla 1.3	Modelos de despliegue en la nube [6].	9
Tabla 1.4	Productos y Servicios en Microsoft Azure.	11
Tabla 1.5	Verbos HTTP usados en REST.	13
Tabla 1.6	Funciones de Android Studio.	16
Tabla 2.1	Requerimientos Funcionales del Módulo Catálogo.	21
Tabla 2.2	Requerimientos Funcionales del Módulo Ventas.	22
Tabla 2.3	Requerimientos Funcionales del Módulo Servicios.	22
Tabla 2.4	Requerimientos Funcionales del Módulo Cursos.	22
Tabla 2.5	Requerimientos Funcionales del Módulo Informativo.	23
Tabla 2.6	Requerimientos Funcionales del Módulo Usuarios.	23
Tabla 2.7	Requerimientos Funcionales del Módulo Móvil.	23
Tabla 2.8	Requerimientos No Funcionales.	24
Tabla 2.9	Características principales de Microsoft Azure [34].	25
Tabla 2.10	Características del servidor B1S en Microsoft Azure [35].	26
Tabla 2.11	Colección "Usuario".	28
Tabla 2.12	Módulos que conforman el prototipo de software distribuido.	30
Tabla 2.13	Detalla de la Clase Carrera.	36
Tabla 2.14	Métodos HTTP asociados a las acciones en la ruta "Product".	50
Tabla 3.1	Requerimientos Funcionales del Módulo Catálogo.	61
Tabla 3.2	Requerimientos Funcionales del Módulo Ventas.	66
Tabla 3.3	Requerimientos Funcionales del Módulo Servicios.	68
Tabla 3.4	Requerimientos Funcionales del Módulo Cursos.	75
Tabla 3.5	Requerimientos Funcionales del Módulo Informativo.	77
Tabla 3.6	Requerimientos Funcionales del Módulo Usuarios.	79
Tabla 3.7	Requerimientos Funcionales del Módulo Móvil.	82
Tabla 3.8	Lista de cumplimiento de los RF.	84
Tabla 3.9	Lista de cumplimiento de los RF (continuación).	85
Tabla 3.10	Requerimientos No Funcionales.	86
Tabla 3.11	Lista de cumplimiento de los RNF.	89

ÍNDICE DE CÓDIGOS

Código 1.1 Levantamiento de un servidor en Node.js.....	14
Código 1.2 Inicio de un servidor con Express.	15
Código 2.1 Comandos para instalar MongoDB.	46
Código 2.2 Comandos para instalar Node.js.....	46
Código 2.3 Comando para instalar Express.....	46
Código 2.4 Comando para instalar Angular.	46
Código 2.5 Código para la creación de la colección "Producto".	49
Código 2.6 Archivo que maneja la ruta al recurso Producto.....	51
Código 2.7 Función " <i>create_product</i> " del controlador " <i>ProductController</i> ".	52
Código 2.8 Plantilla HTML para el " <i>ProductComponent</i> ".....	54
Código 2.9 Método <i>saveProduct()</i> en el componente de clase de " <i>ProductComponent</i> ".	55
Código 2.10 <i>Layout</i> para la interfaz gráfica.	57

RESUMEN

Este Proyecto Técnico propone el diseño e implementación de un prototipo de software distribuido compuesto por una aplicación web y móvil. La aplicación web estará basada en una arquitectura de 3 capas (Presentación, Negocio y Datos) que mejorará la gestión de productos y servicios de la Asociación de Estudiantes de Ingeniería Electrónica (AEIE). Cabe mencionar que, la Capa de Datos, Negocio y Presentación se encuentra ubicado en un proveedor Cloud Computing. La aplicación móvil será programada con el stack MEAN. Se conoce con el acrónimo de MEAN (por sus siglas MongoDB, Express, Angular y Node.js) al *stack* o compilación de estas tecnologías para desarrollar aplicaciones web usando como único lenguaje de programación JavaScript. La aplicación móvil para dispositivos Android estará dirigida a los estudiantes y funcionará como una interfaz de tipo informativo. Este documento está dividido en cuatro capítulos.

El Capítulo 1 detalla los objetivos y alcance que tendrán el Proyecto de Técnico; además describe de forma general conceptos como: servidor web, aplicación móvil y base de datos. Trata los fundamentos teóricos sobre *Cloud Computing*, así como también, los diferentes proveedores que brindan un ambiente de este tipo.

El Capítulo 2 se enfoca en detallar la metodología empleada para el desarrollo de este Proyecto Técnico dividido en dos fases: Diseño e Implementación.

El Capítulo 3 muestra las pruebas realizadas en el prototipo de sistema distribuido para verificar su correcto funcionamiento.

El Capítulo 4 cubre las conclusiones y recomendaciones finalizada la implementación de este Proyecto Técnico.

PALABRAS CLAVE: Computación en la Nube, Sistema distribuido, Android, JavaScript.

ABSTRACT

This Technical Project proposes the design and implementation of a distributed software prototype composed of a web and mobile application. The mobile application will be based on a 3-layer architecture (Presentation, Business and Data) that will improve the management of products and services of Association of Students of Electronic Engineering (AEIE). It is worth mentioning that, Data, Business and Presentation Layer are located in a Cloud Computing provider. The web application will be programmed with the MEAN stack. The acronym of MEAN (MongoDB, Express, Angular and Node.js) is known to the stack or compilation of these technologies to develop web applications using only JavaScript as programming language. The mobile application for Android devices will intended to students and will work as an informative interface. This document is divided into four chapters.

Chapter 1 details the objectives and scope of this Technical Project; In addition, it generally describes concepts such as: web server, mobile application and database. It deals with the theoretical foundations on Cloud Computing, as well as, the different providers that provide such environment.

Chapter 2 focuses on detailing the methodology used for the development of this Technical Project divided into two phases: Design and Implementation.

Chapter 3 shows the tests performed on the distributed system prototype to verify its correct operation.

Chapter 4 covers the conclusions and recommendations after the implementation of this Technical Project.

KEYWORDS: Cloud Computing, Distributed System, Android, JavaScript.

1 INTRODUCCIÓN

Este capítulo, detalla en primera instancia los objetivos y alcance que tendrá el Proyecto de Titulación; además, describe de forma general conceptos como: servidor web, aplicación móvil y base de datos. Trata los fundamentos teóricos sobre *Cloud Computing*, así como también, los diferentes proveedores que brindan un ambiente de este tipo. Por otra parte, describe las herramientas que se utilizarán para el desarrollo del Proyecto en las Capas de Presentación, Lógica de Negocio y Datos. Por último, enuncia la metodología Kanban que será usada para el desarrollo y verificación del cumplimiento correcto del Trabajo de Titulación.

1.1 OBJETIVOS

El objetivo general de este Proyecto Técnico es el de “Desarrollar un prototipo de software distribuido para la gestión de productos y servicios ofertados por la Asociación de Estudiantes de Ingeniería Electrónica”.

Los objetivos específicos del Proyecto Técnico son:

- Analizar los fundamentos teóricos necesarios para el desarrollo del prototipo de sistema distribuido.
- Diseñar los módulos que componen el prototipo de sistema distribuido.
- Implementar los módulos del prototipo de sistema distribuido.
- Analizar los resultados obtenidos en las pruebas realizadas sobre el prototipo.

1.2 ALCANCE

El Trabajo de Titulación propone el diseño e implementación de un prototipo de software distribuido, basado en una arquitectura de 3 capas (Presentación, Negocio y Datos) que mejore la gestión de productos y servicios de la Asociación de Estudiantes de Ingeniería Electrónica (AEIE). Cabe mencionar que, la Capa de Datos, Negocio y Presentación se encontrarán ubicadas en un proveedor *Cloud Computing*. Además, de una aplicación Android dirigida a los estudiantes que implementará una interfaz de tipo informativo. El sistema web será programado con el *stack* MEAN. Se conoce con el acrónimo de MEAN (por sus siglas MongoDB, Express, Angular y Node.js) al *stack* o compilación de estas tecnologías para desarrollar aplicaciones web usando como único lenguaje de programación JavaScript [1].

A continuación, se describen las herramientas que se utilizarán en la implementación del módulo web:

- Para la capa de Datos se implementará una base de datos utilizando MongoDB.
- Se utilizará el *framework*¹ Angular para el desarrollo *frontend* correspondiente a la capa de Presentación.
- Se desarrollará la capa de Negocio utilizando Express, el cual es un *framework* web para Node.js.

Por otra parte, para el módulo móvil se desarrollará la aplicación con Android Studio.

La Figura 1.1 muestra un esquema general del prototipo, donde se visualiza la propuesta basada en la arquitectura de 3 capas.

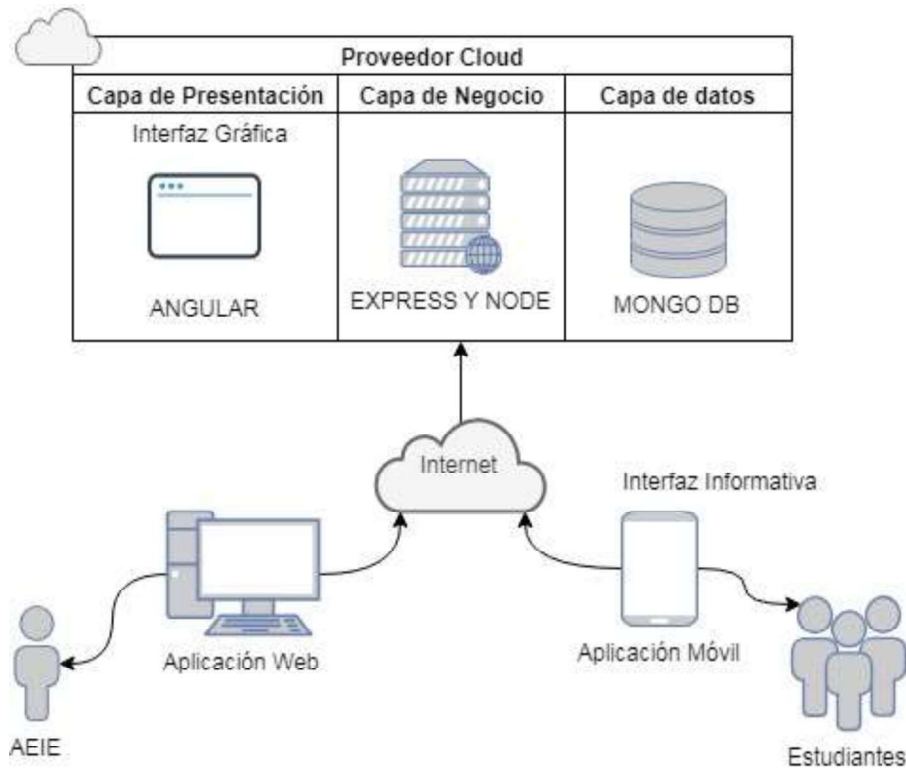


Figura 1.1 Esquema del prototipo de sistema distribuido para la gestión de productos y servicios ofertados por la AEIE.

El sistema web dirigido a la administración de la AEIE, dispondrá de los siguientes módulos que ejecutarán las siguientes acciones:

¹ *framework*: conjunto de librerías y herramientas que facilitan la construcción de una aplicación web.

CATÁLOGO

- Registro de stock de productos: Añadir, modificar y eliminar snacks, bebidas, dispositivos electrónicos.
- Añadir, modificar y eliminar proveedores de productos.

VENTAS

- Añadir y modificar ventas del día.

USUARIOS

- Administrar usuarios. Creación, actualización y eliminación de un usuario.
- Añadir, modificar y eliminar aportantes.

SERVICIOS

- Añadir, modificar y eliminar casilleros para alquiler.
- Añadir o eliminar el registro de un estudiante que alquila un casillero.
- Notificar mediante correo electrónico el estado de alquiler del casillero al estudiante.

CURSOS

- Añadir, modificar y eliminar cursos ofertados por la AEIE.
- Registrar estudiantes en los cursos.

INFORMATIVO

- Publicar eventos, avisos de tipo informativo que serán visualizados en la aplicación móvil destinada a los estudiantes.

La aplicación móvil orientada hacia los estudiantes, realizará las actividades correspondientes:

- Visualizar productos ofertados por la AEIE.
- Visualizar cursos ofertados por la AEIE.
- Visualizar eventos, notificaciones de tipo informativo publicados por la AEIE; como: objetos personales extraviados, promociones, entre otras.
- Visualizar la disponibilidad de casilleros.

Para las notificaciones del casillero, el sistema web enviará un correo electrónico de forma automática a la persona que lo alquila.

Todas las tecnologías y herramientas serán posteriormente estudiadas.

1.3 MARCO TEÓRICO

1.3.1 APLICACIONES MÓVILES Y SMARTPHONES

La sociedad vive un fenómeno de exposición y uso creciente de la tecnología celular en, prácticamente, todas las áreas del quehacer humano. En gran parte, esto sucede por la disminución del costo de los equipos de telefonía celular e incremento en sus capacidades tecnológicas. La tecnología y el cómputo ubicuo propician el desarrollo de un dispositivo portátil que cabe en la palma de la mano, conocido como *smartphone* o teléfono inteligente, cuya aceptación se debe a las características que posee como son: gran capacidad de comunicación, procesamiento y almacenamiento, manejo de contenidos multimedia y fácil integración a redes inalámbricas [2].

A diferencia de los teléfonos móviles tradicionales, los *smartphones* ofrecen más funciones, incluyendo diversos contenidos de Internet y reproductores multimedia. Particularmente, un aspecto innovador de los *smartphones* es que el usuario está en capacidad de descargar varios tipos de aplicaciones móviles o *apps* en su teléfono. Una aplicación móvil es una aplicación de software diseñada para ejecutarse en teléfonos inteligentes y *tablets*. Es así, que la aparición del mercado de aplicaciones ha transformado la industria de la telefonía móvil [3].

1.3.1.1 Análisis del mercado móvil

El mercado de sistemas operativos móviles ha evolucionado con el pasar de los años. Según Gartner², en el primer trimestre del 2018, las principales empresas de dispositivos móviles fueron Samsung, Apple y Huawei, como se observa en la Figura 1.2. La marca Samsung ha abarcado el 20.5% del mercado global, mientras que Apple posee el 10.5%. El sistema operativo Android es el más utilizado por la mayoría de los fabricantes, a diferencia de Apple que posee su propio sistema operativo [4]. La popularidad de Android se debe a que es un sistema de código abierto, contrariamente a iOS que es controlado por Apple, lo que permite a distintos fabricantes integrarlo en sus aparatos sin necesidad de pagar licencia [5].

² Gartner: es la compañía líder de investigación y asesoría del mundo y miembro del SP&500.

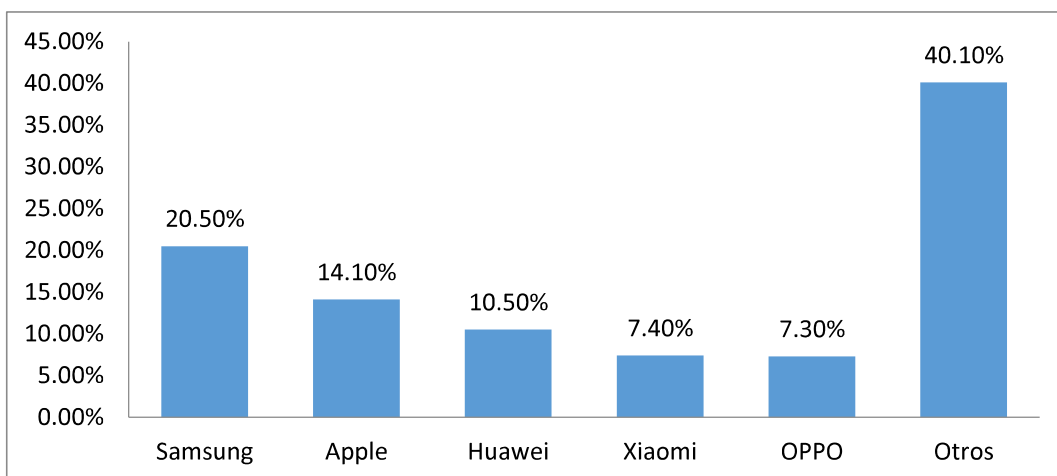


Figura 1.2 Ventas a nivel mundial de teléfonos inteligentes.

Por otra parte, como se observa en la Tabla 1.1, Android e iOS son dos de los gigantes que lideran el mercado de los sistemas operativos móviles con el 85,9% y 14,1% respectivamente.

Tabla 1.1 Ventas mundiales de *smartphones* a usuarios finales por sistema operativo[4].

Sistema Operativo	Unidades en el primer trimestre del 2018	Porcentaje en el Mercado en el primer trimestre del 2018 (%)
Android	329,313.9	85.9
iOS	54,058.9	14.1
Otros	131.1	0.0
Total	383,503.9	100.0

Basado en la Tabla 1.1, se evidencia que el sistema operativo Android está comercializado mayormente a través de las distintas marcas de teléfonos inteligentes, por tal motivo, este Proyecto de Titulación será desarrollado para teléfonos con sistema operativo Android con el propósito de abarcar la mayor cantidad posible de usuarios.

1.3.2 CLOUD COMPUTING

La Computación en la Nube es un modelo que permite el acceso ubicuo, conveniente y bajo demanda de la red a un conjunto compartido de recursos informáticos configurables, por ejemplo: redes, servidores, almacenamiento, aplicaciones y servicios, que pueden ser aprovisionados y lanzados rápidamente con un mínimo esfuerzo de administración o interacción del proveedor de servicios [6]. La industria de la Tecnología de Información (TI) se ha transformado gracias a la Computación en la Nube ya que el modelo de la

nube ofrece a la misma, cinco características esenciales, tres modelos de servicios y cuatro modelos de despliegue [7].

1.3.2.1 Características esenciales

La Computación en la Nube es un paradigma en evolución. Las características que se mencionan a continuación en la Tabla 1.2 han influido en la popularidad de su uso [6]:

Tabla 1.2 Características esenciales del *Cloud Computing*.

Características	Descripción
Auto-servicio bajo de manda	El consumidor o cliente puede proporcionar capacidades de cómputo según sea necesario sin requerir la interacción humana con el proveedor de servicios.
Amplio acceso a la red	Las capacidades de cómputo están disponibles sobre la red y son accedidas a través de mecanismos estándar que promueven el uso de distintas plataformas de cliente como: teléfonos móviles, <i>laptops</i> , estaciones de trabajos, entre otros.
Resource Pooling	Consiste en la combinación de los recursos informáticos del proveedor para servir a múltiples clientes con distintos recursos, físicos o virtuales, asignados dinámicamente y reasignados bajo la demanda del consumidor.
Elasticidad rápida	Las capacidades pueden ser aprovisionadas y liberadas elásticamente, en algunos casos automáticamente, para escalar con mayor rapidez hacia afuera y hacia adentro de acuerdo con la demanda.
Servicio medido	La utilización de los recursos disponibles se puede medir, monitorear, controlar e informar, proporcionando transparencia al proveedor y consumidor.

1.3.2.2 Modelos de servicios

Un cliente puede interactuar de varias maneras con la nube. A través de la web, tres tipos de modelos de servicios han emergido. El primero se conoce como **Software como Servicio (SaaS)** que consiste en contenido (aplicación) entregado en la web, al cual el usuario accede a través de un navegador web. El software puede residir dentro de cualquier modelo de despliegue en la nube [8]. La Figura 1.3 ilustra el modelo SaaS.

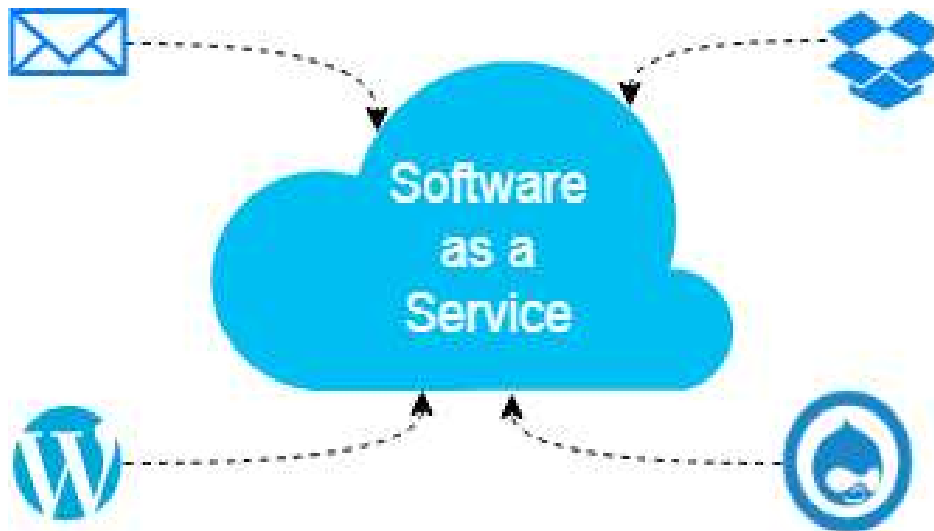


Figura 1.3 Software como Servicio.

Las ventajas que tiene SaaS son: simplicidad de integración, costo y escalabilidad, mientras que la desventaja radica en la percepción de problemas de seguridad [8]. El consumidor no administra o controla la capa de infraestructura incluyendo la red, servidores, sistemas operativos, almacenamiento o incluso las capacidades de la aplicación [6].

El segundo tipo de servicio es **Plataforma como Servicio (PaaS)** que provee una capa tecnológica tal como: servidores, servidores virtuales, sistemas operativos, bases de datos, herramientas de desarrollo, soporte de red para que los desarrolladores desplieguen sus propias soluciones. Es así, que los programadores no necesitan preocuparse de las actualizaciones de *hardware* o sistemas operativos debido a que el proveedor se encarga de estas tareas, permitiendo que los desarrolladores se enfoquen principalmente en sus aplicaciones [8]. En la Figura 1.4 se observa ejemplos de proveedores PaaS como son Cloud9 de Amazon y Heroku.



Figura 1.4 Plataforma como Servicio.

Infraestructura como Servicio (IaaS) es el tercer tipo de servicio en donde el consumidor o cliente es provisto de redes, servidores físicos o virtuales, almacenamiento, procesamiento, y otros recursos computacionales para que el consumidor pueda desplegar y correr software arbitrario. En este tipo de servicio, los desarrolladores o el administrador de sistemas de la compañía deben administrar el *hardware* y software [8]. La Figura 1.5 muestra los principales proveedores de IaaS en el mercado como son Amazon Web Services, Microsoft Azure y Google Cloud Platform.

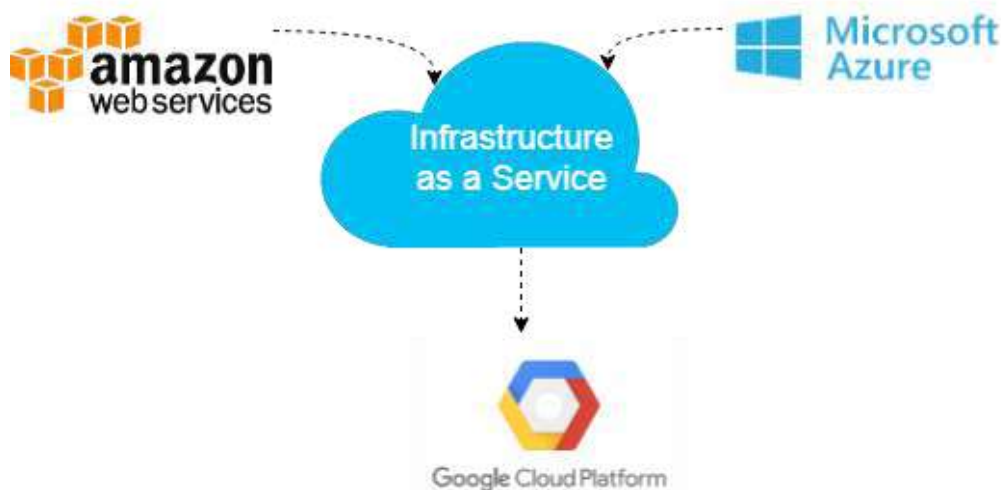


Figura 1.5 Infraestructura como Servicio.

1.3.2.3 Tipos de modelos de despliegue

Dentro de la Computación en la Nube, el modelo de despliegue se encarga de especificar como los recursos son compartidos dentro de la nube. En la Tabla 1.3, se detallan los cuatro modelos de despliegue principales, cada modelo influye en aspectos como escalabilidad, confiabilidad, seguridad y costo [8].

Tabla 1.3 Modelos de despliegue en la nube [6].

Modelo de despliegue	Características	
<p><i>Cloud</i> Privado</p>	<p>La infraestructura en la nube está provista para uso exclusivo por una sola organización que comprende múltiples consumidores (por ejemplo, unidades de negocios).</p>	<p>Puede ser de propiedad, administrado y operado por la organización, un tercero o alguna combinación de ellos, y puede existir dentro o fuera de las instalaciones.</p>
<p><i>Cloud</i> Comunitario</p>	<p>La infraestructura en la nube está aprovisionada para uso exclusivo por una comunidad en específico de consumidores de organizaciones que tienen inquietudes compartidas (por ejemplo, misión, requisitos de seguridad, políticas y consideraciones de cumplimiento).</p>	<p>Puede ser de propiedad, administrada y operada por una o más de las organizaciones de la comunidad, un tercero o alguna combinación de ellas, y puede existir dentro o fuera de las instalaciones.</p>
<p><i>Cloud</i> Público</p>	<p>La infraestructura en la nube está provista para uso abierto por el público en general.</p>	<p>Puede ser propiedad de, administrado y operado por una organización empresarial, académica o gubernamental, o una combinación de ellos. Existe en las instalaciones del proveedor de la nube.</p>
<p><i>Cloud</i> Híbrido</p>	<p>La infraestructura en la nube es una composición de dos o más infraestructuras en la nube distintas (privada, comunitaria o pública).</p>	

1.3.2.4 Proveedores Cloud

Un proveedor de servicios en la nube es una compañía externa que ofrece servicios de plataforma, infraestructura, aplicaciones o almacenamiento basados en la nube. De esta manera, los clientes solo deben pagar por la cantidad de servicios que utilizan, según requiera su negocio [9]. Dentro de las empresas más populares que ofrecen servicios en la nube en el mercado, están Microsoft Azure, Amazon Web Services y Google Cloud Platform. A continuación, se detalla cada una a fin de conocer sus productos ofertados que han permitido a distintas empresas del sector de Tecnologías de Información (TI) optar por una alternativa en la nube.

Microsoft Azure es un conjunto de servicios en la nube para ayudar a una organización a satisfacer sus necesidades comerciales. Otorga la libertad de crear, administrar e implementar aplicaciones en una red mundial enorme con sus herramientas y marcos favoritos [10]. Posee 54 regiones globales, disponible en 140 países. Además, Microsoft Azure es compatible con tecnologías de código abierto, por lo que se puede usar distintas herramientas y tecnologías.

Por otra parte, Amazon Web Services (AWS) es una plataforma segura de servicios en la nube, ofreciendo potencia de cómputo, almacenamiento de bases de datos, entrega de contenido y otras funcionalidades que ayudan a las empresas a escalar y crecer. Abarca 61 zonas de disponibilidad dentro de 20 regiones geográficas alrededor del mundo [11].

Por otro lado, Google Cloud Platform (GCP) es otra de las alternativas disponibles perteneciente a Google, la cual consiste en un conjunto de activos físicos, como computadoras, unidades de disco duro, y recursos virtuales, como máquinas virtuales (VM), que se encuentran en los centros de datos de Google en todo el mundo. Cada ubicación del centro de datos está en una región global. Las regiones incluyen Estados Unidos central, Europa occidental y Asia oriental. Cada región es una colección de zonas, que están aisladas unas de otras dentro de la región [12].

En un mundo *multicloud* dominado principalmente por los 3 proveedores, Amazon Web Services, Google Cloud Platform y Microsoft Azure, cada vez se tiene a disposición mayor cantidad de productos y soluciones de calidad con una disponibilidad excelente, así como también, alta seguridad y rendimiento; permitiendo a las empresas sin importar su tamaño confiar en los servicios en la nube para apoyar sus operaciones en línea. La Tabla 1.4 presenta una comparación entre algunos productos y servicios de los principales proveedores de *Cloud Computing*.

Tabla 1.4 Productos y Servicios en Microsoft Azure.

Productos y Servicios	Amazon Web Services	Microsoft Azure	Google Cloud
Redes y Conectividad	<i>Virtual Private Cloud</i>	<i>Virtual Network</i>	<i>Virtual Private Cloud</i>
	<i>Route 53</i>	<i>Azure DNS</i>	<i>Cloud DNS</i>
Almacenamiento	<i>Simple Storage Service (S3)</i>	<i>Storage</i>	<i>Cloud Storage</i>
Recursos Informáticos	<i>Elastic Compute Cloud (EC2)</i>	<i>Virtual machines</i>	<i>Compute Engine</i>
Bases de datos relacionales	<i>RDS</i>	<i>SQL Database</i>	<i>Cloud SQL</i>
Bases de datos NoSQL	<i>DynamoDB</i>	<i>Azure Cosmos DB</i>	<i>Cloud Bigtable</i>

1.3.3 DESCRIPCIÓN DE HERRAMIENTAS Y TECNOLOGÍAS POR USAR EN LA CAPA DE DATOS

La Capa de Datos corresponde a la gestión del almacenamiento de la información dentro de la base de datos.

1.3.3.1 MongoDB

Una base de datos es un conjunto de información almacenada sin redundancias innecesarias en un soporte informático y accesible simultáneamente por distintos usuarios y aplicaciones [13]. Las bases de datos representan uno de los grandes aportes a la informática, ya que, han solucionado problemas en los sistemas de archivos tradicionales como: redundancia e inconsistencia de datos, dificultad en el acceso a datos, aislamiento de datos, problemas de seguridad, entre otros [14].

MongoDB es una base de datos *open-source* documental con gran escalabilidad y flexibilidad. Trabaja con el concepto de “colección” y “documento”, almacenando los datos en documentos BSON. BSON es una representación binaria de documentos JSON³, lo

³ JSON (*JavaScript Object Notation*): formato de texto simple, legible para los humanos, para expresar datos estructurados con soporte en muchos lenguajes de programación.

que significa que los campos pueden variar de un documento a otro y la estructura de los datos se puede cambiar con el tiempo [15].

Una “colección” es un grupo de documentos de MongoDB, que no está forzada a tener un esquema. Por otra parte, un “documento” es un conjunto de pares clave-valor que tiene una estructura dinámica, es decir, los documentos en una misma colección no necesitan tener el mismo conjunto de campos. Un modelo de datos basados en documentos puede representar estructuras de datos ricas y jerárquicas [16].

1.3.3.2 Mongoose

Mongoose es una biblioteca de Modelado de Datos de Objetos (ODM) que proporciona un entorno de modelado riguroso para los datos, aplicando la estructura según sea necesario y manteniendo la flexibilidad que hace que MongoDB sea poderoso [17]. Es de código abierto con licencia MIT⁴ y también es mantenido por MongoDB, Inc. La principal característica de Mongoose es que puede definir esquemas para las colecciones que usará la base de datos en MongoDB. Mongoose también tiene utilidades para simplificar los patrones de devolución de llamada de Node.js como pueden ser consultas o registro de datos, de tal manera que la manipulación de datos sea más fácil. En general, Mongoose hace que sea aún más fácil usar MongoDB junto con Node.js [18].

1.3.4 DESCRIPCIÓN DE HERRAMIENTAS Y TECNOLOGÍAS POR USAR EN LA CAPA DE LÓGICA DE NEGOCIO

La Capa de Lógica de Negocio está relacionada con el procesamiento de la información, actuando como puente entre la Capa de Presentación y de Datos.

1.3.4.1 Servidor Web y Servicios Web

Un servidor web es una computadora que almacena y entrega contenido desde un sitio web a los clientes que lo soliciten. El contenido entregado puede ser variado, teniendo así: texto, imágenes, video, datos de aplicación, entre otros. Este contenido puede ser entregado a los clientes mediante *HyperText Transfer Protocol* (HTTP), el cual permite la transferencia de información a través de internet [19].

Un servicio web es una interfaz, accesible por protocolos estándar usados en Internet, que permite acceder a las funcionalidades de un objeto concreto, sin importar las tecnologías ni plataformas implicadas en la petición. Existen dos tipos de servicios web

⁴ MIT: licencia de software que fue creada en el Instituto Tecnológico de Massachusetts y es posible usarla tanto para licenciar software libre como software no libre.

muy utilizados que son SOAP y REST. El primero conocido como Protocolo de Acceso de Objeto Simple (SOAP), es un protocolo de empaquetamiento estandarizado para los mensajes compartidos por las aplicaciones basado en XML⁵ [20].

Por otra parte, REST significa Transferencia de Estado Representacional, que es una arquitectura de software, que indica como los datos pueden ser transferidos elegantemente. En REST, cuatro de los verbos de HTTP son usados para proveer una funcionalidad de tipo CRUD (*Create, Read, Update, Delete*) y son: POST, GET, PUT y DELETE. Una de las claves principales de REST es el concepto de recurso, donde cada recurso tiene asociado una URI⁶ con el mismo [21].

Los servicios de tipo REST se ocupan de la transferencia de representaciones de recursos. Una representación puede ser JSON o XML [22]. La Tabla 1.5 resume los verbos HTTP y como los mismos son aplicados en los servicios web de tipo REST.

Tabla 1.5 Verbos HTTP usados en REST.

Verbo	Descripción
GET	Recupera un recurso identificado por una URI.
POST	Envía un recurso al servidor para ser almacenado en la ubicación identificada por la URI.
PUT	Envía un recurso al servidor. Actualiza el recurso en la ubicación identificada por la URI.
DELETE	Elimina un recurso identificado por la URI.

1.3.4.2 Node.js

Node.js es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome. V8 es el motor de JavaScript y *WebAssembly* de alto rendimiento y código abierto de Google, está escrito en C ++ y es utilizado en Chrome así como también en Node.js, entre otros. Node.js es multiplataforma, ya que está disponible para Windows, Linux y Mac. Node.js utiliza un modelo basado en eventos no bloqueante, que lo hace ligero y eficiente [23].

⁵ XML (*Extensible Markup Language*): lenguaje que permite definir un conjunto de reglas para codificar documentos en un formato que sea legible para humanos y máquinas

⁶ URI (*Uniform Resource Identifier*): es un conjunto de caracteres que identifican a un recurso en particular.

Node.js puede manejar múltiples conexiones de red concurrentes con poca sobrecarga, que lo hace ideal para aplicaciones de tiempo real con uso intensivo de datos. Permite usar JavaScript en el lado del servidor, es así que se puede construir diferentes tipos de aplicaciones en red, por ejemplo, servicios de aplicaciones web, API, HTTP *proxy* entre otros [24].

El Código 1.1, muestra un ejemplo donde se levanta un servidor con Node.js, que responde con el texto “*Hola Mundo*”, cada vez que se haga una petición HTTP a la dirección “127.0.0.1”. En la línea 1, se utiliza el módulo “*http*” que permite manejar las solicitudes de tipo HTTP que llegan al servidor. Por otra parte, en las líneas 3 y 4 se define el puerto y dirección IP del servidor. Entre las líneas 6 al 10, se define la acción que ejecuta el servidor, es decir, la respuesta del texto mencionado anteriormente. Finalmente en la línea 12, se coloca en modo de escucha al servidor en espera de alguna solicitud HTTP. Este ejemplo, busca mostrar la sencillez que ofrece Node.js para escribir código del lado del servidor mediante JavaScript.

```
1  const http = require('http');
2
3  const hostname = '127.0.0.1';
4  const port = 3000;
5
6  const server = http.createServer((req, res) => {
7    res.statusCode = 200;
8    res.setHeader('Content-Type', 'text/plain');
9    res.end('Hola Mundo\n');
10 });
11
12 server.listen(port, hostname, () => {
13   console.log(`El servidor se está ejecutando en http://\${hostname}:\${port}/`);
14 });
```

Código 1.1 Levantamiento de un servidor en Node.js.

1.3.4.3 Express

Express es un *framework* de aplicaciones web para Node.js, mínimo y flexible que proporciona un conjunto sólido de características para las aplicaciones web y móviles. En el Código 1.2, se observa el inicio de un servidor con el *framework* Express, donde la aplicación escucha las conexiones en el puerto 3000. La aplicación responde con “*Hello World!*” a las solicitudes a la ruta raíz. Para cada ruta de acceso diferente, responderá con un código de error (404). Este código de estado HTTP, indica que la solicitud ha encontrado el servidor pero el recurso solicitado no existe. Comparando el Código 1.2 con el Código 1.1, se puede visualizar la sencillez que brinda un *framework* para escribir aplicaciones web simplificando la manera de escribir código.

```

1  var express = require('express');
2  var app = express();
3
4  app.get('/', function (req, res) {
5    res.send('Hello World!');
6  });
7
8  app.listen(3000, function () {
9    console.log('Example app listening on port 3000!');
10 });

```

Código 1.2 Inicio de un servidor con Express.

1.3.4.4 SendGrid

Es un servicio de correo electrónico basado en la nube, desarrollado para resolver los desafíos de entregar correos electrónicos de manera confiable a nombre de empresas en crecimiento. Dependiendo de la necesidad su precio puede variar, desde una cuenta gratuita con capacidad de envío de 40,000 correos durante el mes hasta un estimado de 1'500,000 correos por mes con un valor aproximado de \$699.95 dólares [25].

Sendgrid está especialmente pensado en los desarrolladores de software, es por eso que dispone de varias librerías que soportan lenguajes de programación como: PHP, Java, Node.js, Go, C#, Ruby y Python.

1.3.4.5 OneSignal

OneSignal es un servicio de notificaciones *push* confiable y de gran volumen para sitios web y aplicaciones móviles. Ofrece soporte para la mayoría de plataformas nativas móviles principales proporcionando un *kit* de desarrollo de software (SDK) dedicado para cada plataforma, una API RESTful para interactuar con aplicaciones del lado del servidor personalizadas por el usuario y un panel en línea para que especialistas en marketing diseñen y envíen notificaciones *push* [26].

1.3.5 DESCRIPCIÓN DE HERRAMIENTAS Y TECNOLOGÍAS POR USAR EN LA CAPA DE PRESENTACIÓN

La capa de Presentación incluye la presentación de la información al usuario y la interacción con él.

1.3.5.1 Angular

Angular es una plataforma y *framework* para la construcción de aplicaciones web basadas en HTML y TypeScript⁷. Angular combina plantillas declarativas, inyección de dependencia, herramientas de extremo a extremo y mejores prácticas integradas para resolver los desafíos del desarrollo [27]. Dentro de la arquitectura de Angular cabe mencionar sus principales elementos que son módulos, componentes, plantillas directivas, enlace de datos, servicios, inyección de dependencia, y enrutamiento.

1.3.5.2 Visual Studio Code

Visual Studio Code es un editor de código fuente ligero, pero potente que se ejecuta en un computador de escritorio y está disponible para Windows, macOS y Linux. Viene con soporte incorporado para JavaScript, TypeScript y Node.js. Además, tiene un rico ecosistema de extensiones para otros lenguajes de programación (como C ++, C #, Java, Python, PHP, Go) y tiempos de ejecución (como .NET y Unity) [28].

1.3.5.3 Android Studio

Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android y se basa en IntelliJ IDEA. Está disponible para ser instalado en Windows, Linux, y Mac. Android Studio ofrece funciones que aumentan la productividad durante la compilación de *apps* para Android mostradas en la Tabla 1.6 [29]:

Tabla 1.6 Funciones de Android Studio.

Funciones durante la compilación en Android Studio	Un sistema de compilación basado en Gradle flexible.
	Un emulador rápido con varias funciones.
	Un entorno unificado en el que se puede realizar desarrollos para todos los dispositivos Android.
	<i>Instant Run</i> para aplicar cambios mientras una <i>app</i> se ejecuta sin la necesidad de compilar un nuevo APK.
	Gran cantidad de herramientas y <i>frameworks</i> de prueba.
	Herramientas Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versión, etc.
	Soporte incorporado para Google Cloud Platform, lo que facilita la integración de <i>Google Cloud Messaging</i> y <i>App Engine</i> .

⁷ Typescript: *superset* de JavaScript, desarrollado por Microsoft, que compila a JavaScript puro.

1.3.6 METODOLOGÍA ÁGIL DE DESARROLLO KANBAN

Kanban es una técnica simple de administración de proyectos basada en el mecanismo de planificación creado por Toyota. El uso de Kanban permite enfocar el tiempo y la energía en entregar valor a los clientes [30]. En el campo del desarrollo de software, Kanban se aplica a través de un tablero. La Figura 1.6 muestra un ejemplo de tablero Kanban aplicado al software.

Las tareas son escritas en notas, de tal manera que se mueven en dirección de izquierda a derecha. Es importante mencionar que, una tarea no puede pasar a la siguiente columna mientras no sea terminada por la persona o equipo encargado de la misma. Cada columna tiene un nombre a fin de ilustrar donde se encuentra cada ítem o tarea en el flujo de trabajo.

El concepto del tablero Kanban es usado para colocar un límite visible al número de tareas permitidas en cualquier columna, de tal forma que evita los cuellos de botella. Además, optimiza el proceso para que el tiempo de entrega sea lo más pequeño y predecible posible [31].

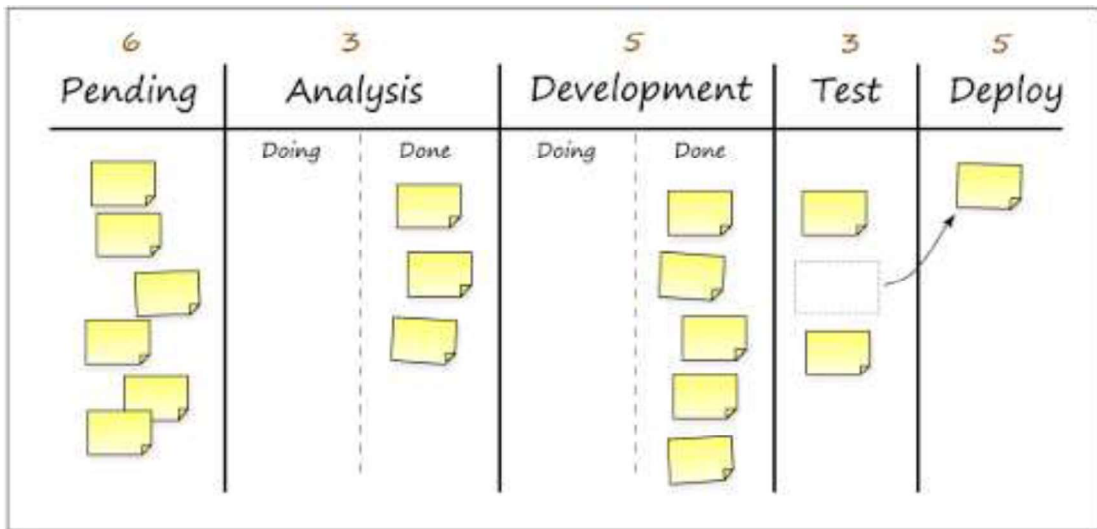


Figura 1.6 Tablero Kanban para Desarrollo de Software.

Visualizar el flujo de trabajo es un principio clave en Kanban. Es así, que manipular las notas en el tablero es más atractivo, rápido y simple que hacerlo electrónicamente. Sin embargo, en el caso de organizaciones de gran tamaño el uso de herramientas en línea ayudan a que el administrador de proyectos pueda transferir rápidamente el estado del trabajo en progreso [30].

2 METODOLOGÍA

En este capítulo, en primera instancia se presenta la situación actual de los productos y servicios ofertados por la Asociación de Estudiantes de Ingeniería Electrónica (AEIE). Luego, muestra el diseño y posteriormente la implementación de los componentes que constituyen el prototipo de Sistema Distribuido.

En el Diseño se definen los Requerimientos Funcionales y No Funcionales con los que cumplirá el prototipo, contribuyendo con el comportamiento del sistema; mientras que, el apartado de Implementación está basado en la codificación del diseño. Los componentes que forman parte del prototipo serán separados en 3 Capas: Datos, Lógica de Negocio y Presentación.

2.1 SITUACIÓN ACTUAL DE LOS PRODUCTOS Y SERVICIOS OFERTADOS POR LA ASOCIACIÓN DE ESTUDIANTES DE INGENIERÍA ELECTRÓNICA (AEIE)

En la actualidad, la Asociación de Estudiantes de Ingeniería Electrónica (AEIE) de la Escuela Politécnica Nacional presenta un problema en la gestión de sus productos y servicios ofertados, como: registro de ventas de snacks, bebidas y dispositivos electrónicos; registro de proveedores; y notificación de estado de alquiler de casilleros. El problema radica en las actividades realizadas de forma manual, así como también, en una gran cantidad de información almacenada en hojas de datos de Microsoft Excel, por ejemplo: registro de aportantes, alquiler de casilleros, *stock* de productos e inscripción de cursos.

Además, cabe resaltar que, la notificación del estado de alquiler de casillero se realiza colocando una nota escrita pegada en la puerta del casillero. La notificación tiene el propósito de indicar al estudiante que el alquiler de su casillero ha terminado, de tal manera que debe acercarse a la AEIE a retirar sus pertenencias del casillero. Por consiguiente, se evidencia un método de notificación deficiente de los servicios ofrecidos por la AEIE.

Con el fin de mejorar la gestión de productos y servicios ofertados, se plantea el diseño e implementación de un prototipo de software distribuido que permita a la AEIE desempeñar su labor de una mejor manera. La AEIE se beneficiará a través del desarrollo de este trabajo, ya que, el software propuesto automatizará procesos que se realizan de forma manual; previniendo la pérdida de documentos y ahorrando espacio

físico. Por otra parte, brindará eficiencia en la notificación de alquiler de casilleros a través de correo electrónico.

Además, con el fin de permitir la interacción entre ambas partes (estudiantes - AEIE) se proveerá de una aplicación Android dirigida a los estudiantes con una interfaz de tipo informativo.

2.2 DISEÑO

En este Apartado se detalla el diseño del prototipo de sistema distribuido para la gestión de productos y servicios ofertados por la “AEIE”. En primera instancia, se aplicará la metodología Kanban, para lo cual se definirá el tablero de actividades que permitirá el desarrollo de este trabajo.

A continuación, se definirán los Requerimientos Funcionales y No Funcionales que determinarán el comportamiento del sistema. Además, se realizará la selección del proveedor *Cloud Computing* donde se hospedarán el sistema web. En la Capa de Datos se realiza un diagrama de colecciones siguiendo las buenas prácticas definidas en la documentación oficial de MongoDB. Mientras que, la Capa de Lógica de Negocio incluye el diagrama de casos de uso y diagrama de clases. Finalmente, en la Capa de Presentación se muestran los bosquejos de la aplicación.

2.2.1 TABLERO DE ACTIVIDADES KANBAN

Trello [32], es una aplicación web para administración de proyectos basada en la metodología Kanban. Está disponible para diferentes plataformas: Web, Android y iOS. Además, es de servicio gratuito de manera permanente, evitando así las limitaciones por versión de prueba. Brinda una potente gestión visual mediante tarjetas, listas y tableros, usando la modalidad *drag & drop* (arrastar y soltar).

Por lo tanto, aprovechando los beneficios y características del software Trello, se realizó el Tablero Kanban de actividades correspondiente a la fase de diseño del prototipo. En la Figura 2.1 se observa el tablero Kanban presentando las siguientes columnas: “LISTA DE TAREAS” que se refiere al conjunto de actividades por realizar, correspondiente a la fase de implementación y pruebas de funcionamiento. La lista “TAREAS EN PROCESO” abarca las actividades que se encuentran en proceso de desarrollo, con respecto a la fase de diseño del prototipo. Mientras que, la lista “TAREAS REALIZADAS” corresponde a las actividades completadas.

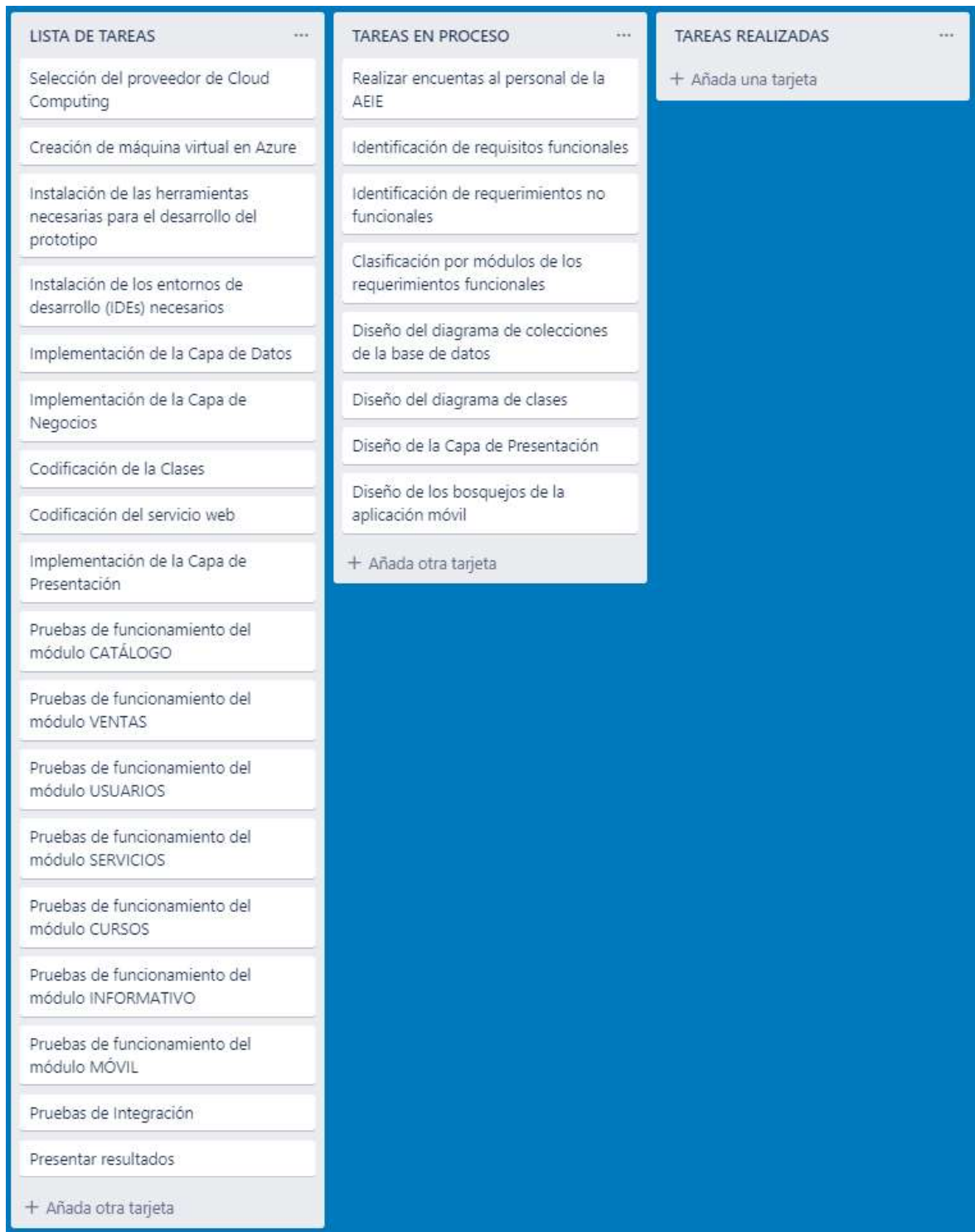


Figura 2.1 Tablero Kanban para la fase de Diseño.

2.2.2 ANÁLISIS DE REQUERIMIENTOS

2.2.2.1 Entrevistas

Para la identificación de los requerimientos del prototipo se ha realizado encuestas al personal de la AEIE, las mismas que se encuentran en el **ANEXO A** junto con las

respuestas. Las preguntas fueron dirigidas al personal administrativo de la AEIE, teniendo como prioridad a las secretarías, quienes están encargadas de la oferta de productos y servicios de la asociación. En este punto se utilizó preguntas abiertas y cerradas, a fin de obtener la información necesaria para diseñar el software. De este modo, se expusieron las necesidades de tal forma que el prototipo propuesto cumpla con los objetivos propuestos.

2.2.2.2 Requerimientos Funcionales

Los Requerimientos Funcionales (RF) son indicadores explícitos que describen el comportamiento que debe tener una solución de software. De acuerdo con las encuestas realizadas y en conjunto con el Plan de Titulación Aprobado, se definen los RF que señalan las funciones del sistema de software.

Con el fin de brindar una solución de software organizada, se ha agrupado los RF por afinidad dando origen a módulos que integrarán el prototipo de sistema de gestión.

Los módulos que permitirán el desempeño de las actividades de la AEIE basado en el prototipo de software distribuido son: Módulo Catálogo, Módulo Ventas, Módulo Servicios, Módulo Cursos, Módulo Informativo, Módulo Usuarios y Módulo Móvil.

En la Tabla 2.1, se observa los RF pertenecientes al **Módulo Catálogo**.

Tabla 2.1 Requerimientos Funcionales del Módulo Catálogo.

RF	DESCRIPCIÓN
RF01	Añadir categoría de productos.
RF02	Modificar categoría de productos.
RF03	Eliminar categoría de productos.
RF04	Añadir productos.
RF05	Modificar productos.
RF06	Eliminar productos.
RF07	Añadir proveedor de productos.
RF08	Modificar proveedor productos.
RF09	Eliminar proveedor de productos.

Para el manejo de las ventas se ha diseñado el **Módulo Ventas**, que abarca las siguientes tareas definidas en la Tabla 2.2.

Tabla 2.2 Requerimientos Funcionales del Módulo Ventas.

RF	DESCRIPCIÓN
RF10	Registrar ventas del día.
RF11	Listar todas las ventas.

Por otra parte, el **Módulo Servicios** describe los siguientes RF detallados en la Tabla 2.3.

Tabla 2.3 Requerimientos Funcionales del Módulo Servicios.

RF	DESCRIPCIÓN
RF12	Registrar semestre.
RF13	Añadir estudiantes.
RF14	Modificar estudiantes.
RF15	Eliminar estudiantes.
RF16	Registrar el aporte de un estudiante.
RF17	Eliminar el aporte de un estudiante.
RF18	Añadir casilleros.
RF19	Modificar casilleros.
RF20	Eliminar casilleros.
RF21	Añadir el alquiler de casillero por un estudiante.
RF22	Eliminar el registro de un estudiante que alquila un casillero.
RF23	Notificar mediante correo electrónico el estado de alquiler del casillero al estudiante.

La Tabla 2.4 muestra las acciones asociadas al **Módulo Cursos**.

Tabla 2.4 Requerimientos Funcionales del Módulo Cursos.

RF	DESCRIPCIÓN
RF24	Añadir cursos ofertados por la AEIE.
RF25	Modificar cursos ofertados por la AEIE.
RF26	Eliminar cursos ofertados por la AEIE.
RF27	Registrar estudiantes en los cursos.

Con relación al **Módulo Informativo**, sus RF se detallan en la Tabla 2.5. Es importante comentar que dicho módulo aporta a la interactividad con el **Módulo Móvil**.

Tabla 2.5 Requerimientos Funcionales del Módulo Informativo.

RF	DESCRIPCIÓN
RF28	Publicar eventos, avisos de tipo informativo que serán visualizados en la aplicación móvil destinada a los estudiantes.
RF29	Modificar avisos publicados.
RF30	Eliminar avisos publicados.

Además, el **Módulo Usuarios** señala en la Tabla 2.6 los requisitos que permitirá al personal de la AEIE el ingreso hacia el sistema web y su correspondiente administración.

Tabla 2.6 Requerimientos Funcionales del Módulo Usuarios.

RF	DESCRIPCIÓN
RF31	Registrar usuario.
RF32	Modificar usuario.
RF33	Eliminar usuario.
RF34	Listar usuarios.

Finalmente, el **Módulo Móvil** permitirá ejecutar las siguientes acciones descritas en la Tabla 2.7. Se debe tener en cuenta que este módulo corresponde a la aplicación móvil destinada a los estudiantes, mientras que los otros módulos mencionados anteriormente forman parte del sistema web dirigido a la administración por parte de la AEIE.

Tabla 2.7 Requerimientos Funcionales del Módulo Móvil.

RF	DESCRIPCION
RF35	Visualizar productos ofertados por la AEIE.
RF36	Visualizar cursos ofertados por la AEIE.
RF37	Visualizar eventos, notificaciones de tipo informativo publicados por la AEIE.
RF38	Visualizar la disponibilidad de casilleros.

2.2.2.3 Requerimientos No Funcionales

Los Requerimientos No Funcionales (RNF) son aquellos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes o cualidades que el producto debe tener como: fiabilidad, tiempo de respuesta, capacidad de almacenamiento, entre otros. En la Tabla 2.8 se observa los RNF con una descripción de los mismos.

Tabla 2.8 Requerimientos No Funcionales.

	Requerimiento No Funcional	Descripción
Usabilidad	RNF01	El sistema debe brindar a los usuarios una interfaz gráfica amigable y de uso sencillo.
Escalabilidad	RNF02	El prototipo debe ser construido en una infraestructura que permita el incremento de recursos informáticos como memoria RAM y almacenamiento.
Disponibilidad	RNF03	El sistema debe estar disponible las 24 horas del día.
Autenticación	RNF04	El acceso al sistema web debe estar restringido por un correo electrónico y contraseña para cada usuario.

2.2.3 SELECCIÓN DEL PROVEEDOR CLOUD COMPUTING

En el **Apartado 1.3.2.4**, se ha analizado los principales proveedores: Google Cloud Platform, Amazon Web Services, y Microsoft Azure. Si bien los nombres de los productos y servicios cambian de un proveedor a otro, estos se enfocan en dar soluciones similares en temas importantes para las empresas como: redes y conectividad, almacenamiento, recursos informáticos, bases de datos, entre otros. El prototipo utilizará herramientas de software libre que se instalarán para su funcionamiento, por tanto, es necesario escoger un proveedor que permita acondicionar el servidor para cumplir con los fines deseados.

Amazon Web Services ofrece una capa gratuita (*Free Tier*) para prueba de sus servicios. De igual manera, Google Cloud Platform permite que los usuarios tengan acceso a su nivel gratuito. Ambos proveedores conceden 12 meses gratuitos para probar sus servicios. Sin embargo para poder abrir dichas cuentas y poder registrarse, se requiere

de una tarjeta de crédito. Esto se vuelve un limitante para estudiantes y jóvenes que no disponen este tipo de tarjeta. Por otra parte, es importante mencionar que Microsoft Azure, con el afán de apoyar a la educación de estudiantes de carreras de TI, ofrece un período gratuito de prueba correspondiente a 12 meses de ciertos productos y servicios localizados en su potente nube. Todo esto sin la necesidad de una tarjeta de crédito y simplemente se debe verificar la condición de estudiante a través de un correo institucional.

Además, según el Cuadrante Mágico de Gartner de IaaS en la nube, para el primer período del 2018, Microsoft Azure recibió la distinción de ser uno de los tres líderes en el mercado de los proveedores de *Cloud Computing*. La extensa evaluación de Gartner de proveedores de IaaS abarca los puntos fuertes de los proveedores; además, las precauciones que se debe tomar y los usos recomendados de los principales proveedores de servicios en el *Cloud* [33]. En la Tabla 2.9 se muestra las características principales de Microsoft Azure.

Tabla 2.9 Características principales de Microsoft Azure [34].

Característica	Descripción
Autoservicio bajo demanda	Los usuarios pueden proveerse de cómputo en la nube sin requerir interacción humana o con el mismo proveedor.
Acceso ubicuo a la red	Todo recurso que se necesite se encuentra en la red y accesible desde la red mediante distintos dispositivos.
Agrupación de recursos independientes de la posición	Los recursos del proveedor se encuentran geográficamente agrupados para servir a múltiples consumidores de manera distribuida y bajo demanda.
Elasticidad rápida	Las funcionalidades se proporcionan de manera muy rápida, incluso puede ser configurable para que crezca dependiendo el ambiente actual.
Servicio medido	El uso de todos los recursos se puede monitorizar, lo que proporciona transparencia tanto al que expone los servicios (el proveedor) como a los que acceden a ellos (los consumidores).
Pago por uso	Sólo se paga por el tiempo que el servicio adquirido está activo o en uso.

Por todo lo anteriormente citado, Microsoft Azure ha sido elegido como proveedor para hospedar el prototipo de software distribuido del Proyecto de Titulación. Dentro de este proveedor *Cloud* se implementará el prototipo de sistema distribuido en un servidor de Tipo B1S, cuyas características se encuentran detalladas en la Tabla 2.10.

Tabla 2.10 Características del servidor B1S en Microsoft Azure [35].

Instancia	B1S
CPU virtual	1
RAM	1 GB
Almacenamiento	4GB
Sistema Operativo	Linux
Horas de cómputo gratis disponible al mes	750 horas

2.2.4 DISEÑO DE LA CAPA DE DATOS

Este apartado inicia con el diseño del diagrama de colecciones basado en los requerimientos de usuarios establecidos previamente. Debido a que MongoDB es una base de datos NoSql, la representación de las tablas en el modelo relacional está dada por colecciones anteriormente definidas en el **Apartado 1.3.3.1**.

2.2.4.1 Diagrama de colecciones

En la Figura 2.2 se aprecia el diagrama de colecciones para la base de datos NoSql, que permitirán el almacenamiento de datos para el prototipo de software distribuido. Este diagrama fue generado utilizando el software *DbSchema*, el cual permite realizar diagramas para MongoDB. *DbSchema* [36] descubre la estructura de una base de datos de tipo MongoDB y la muestra como un diagrama visual. Además, posee características integradas como: generador de consultas visuales, editor de consultas, búsqueda de datos relacionales, generador de datos aleatorios, documentación HTML5 y más.

En la Figura 2.2 se puede observar la colección “*Producto*”, que posibilita el registro de un nuevo producto. Esta colección tiene una relación con las colecciones “*Proveedor*” y “*Categoría*” permitiendo saber quién es el proveedor del producto y a qué categoría pertenece. La colección “*Venta*” permite registrar la información correspondiente a ventas del día, que contiene un documento embebido “*Producto*”, con referencia a la colección “*Producto*” e información sobre un producto vendido y su cantidad. Por otra

parte, para saber quién realizó una venta se tiene la colección “Usuario”, la cual permitirá el acceso al sistema una vez que se haya registrado un usuario.

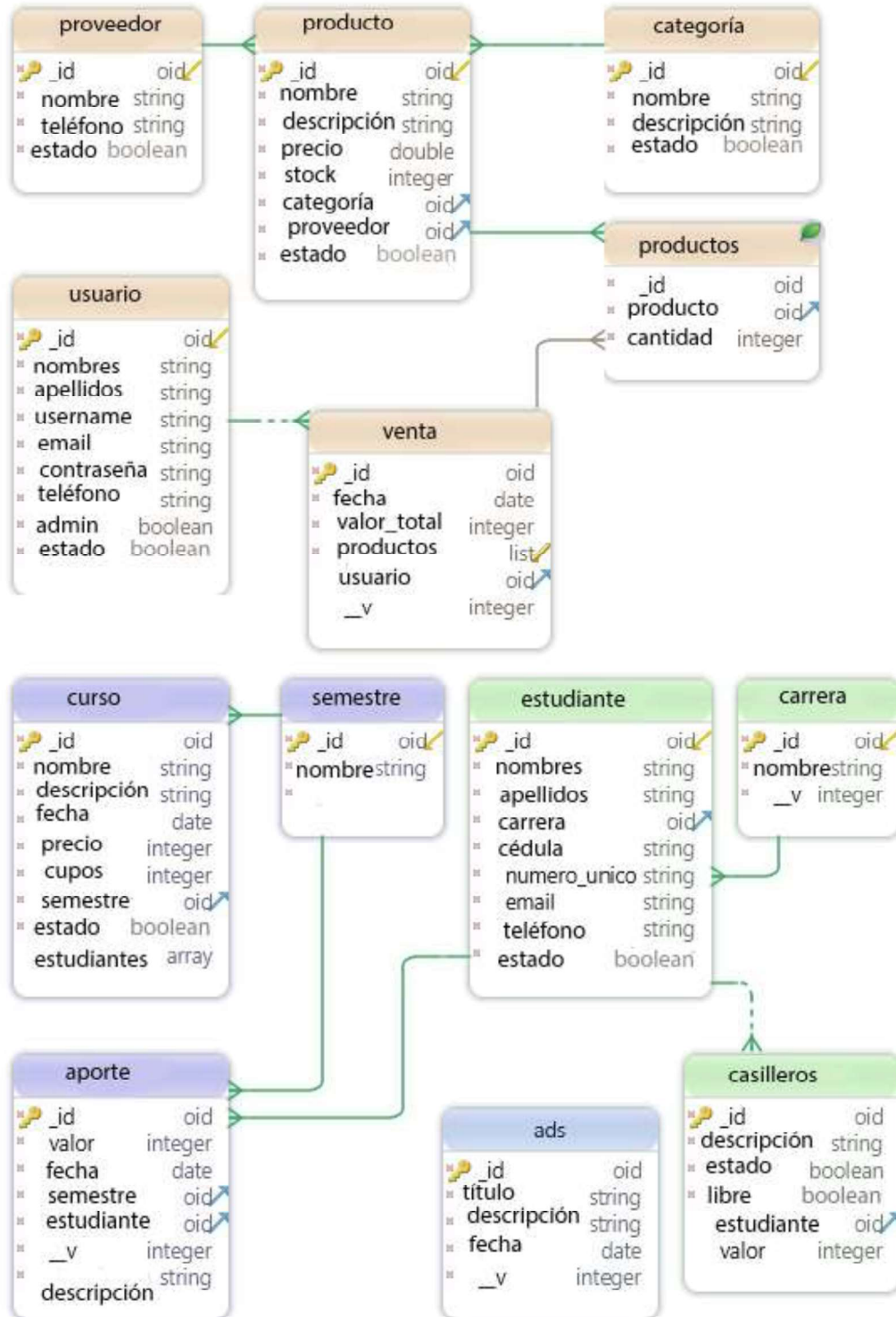


Figura 2.2 Diagrama de colecciones para la base de datos.

La información de un estudiante es almacenada en la colección “*Estudiante*” y para registrar la carrera a la que pertenece se utilizará la colección “*Carrera*”. Por otro lado, en vista de que un estudiante podrá: alquilar un casillero, registrarse en un curso o pagar un aporte, se guardará la información correspondiente a dichas acciones en las colecciones “*Casillero*”, “*Curso*”, “*Aporte*”, respectivamente. Finalmente, la colección “*Ads*” permitirá guardar los avisos de tipo informativo que serán publicados por la AEIE.

En la Tabla 2.11 se explica a modo de ejemplo los campos que forman la colección “*Usuario*”, las colecciones restantes se encuentran en el **ANEXO B**.

Tabla 2.11 Colección "Usuario".

Colección Usuario		
Campo	Tipo de dato	Descripción
_id	<i>String</i>	Identificador único de un usuario.
Nombres	<i>String</i>	Nombres del usuario.
Apellidos	<i>String</i>	Apellidos del usuario.
Username	<i>String</i>	Nombre de un usuario.
Email	<i>String</i>	Correo electrónico de un usuario.
Contraseña	<i>String</i>	Contraseña cifrada del usuario.
Teléfono	<i>String</i>	Teléfono del usuario
Admin	<i>Boolean</i>	Indicador para determinar si un usuario puede tener el rol de Administrador.
Estado	<i>Boolean</i>	Indicador para ocultar el usuario cuando se elimine.

2.2.5 DISEÑO DE LA CAPA DE LÓGICA DE NEGOCIOS

Esta capa presenta la lógica de los procesos que posibilitarán la implementación del prototipo, entre los cuales están los diagramas de Casos de Uso y el Diagrama de Clases.

2.2.5.1 Diagramas de Casos de Uso

Un diagrama de Casos de Uso representa las interacciones entre el sistema, los sistemas externos y los usuarios. En otras palabras, describe de forma gráfica quién utiliza el sistema y la manera en que los usuarios interactúan con el sistema [37]. Es una herramienta valiosa dado que es una técnica de aciertos y errores para obtener los requerimientos del sistema, justamente desde el punto de vista del usuario.

En el presente prototipo se definen tres tipos de actores: Usuario, Administrador y Estudiante. A continuación se detalla el rol de cada uno:

- **Usuario:** este actor hace referencia al personal de la AEIE que administra el sistema web y podrá acceder al mismo sólo si ha sido creado previamente por un Administrador. Un Usuario tiene acceso a los siguientes módulos: Catálogo, Ventas, Servicios, Cursos e Informativo.
- **Administrador:** este actor hace referencia a un Usuario al que se le ha otorgado el rol de Administrador. Un Administrador se encarga de crear Usuarios que puedan iniciar sesión en el sistema web. Un Administrador tiene acceso a todos los módulos del sistema web y estos son: Catálogo, Ventas, Servicios, Cursos, Informativo y Usuarios.
- **Estudiante:** este actor hace referencia a los estudiantes quienes interactuarán exclusivamente con el módulo Móvil. Este módulo corresponde a la aplicación móvil Android dentro del prototipo de software distribuido.

En la Figura 2.3 se presenta los tres actores principales que interactuarán con el prototipo de sistema distribuido. El actor “*Estudiante*” interactuará únicamente con la aplicación móvil Android, mientras que los actores “*Usuario*” y “*Administrador*” interactuarán con el sistema web.

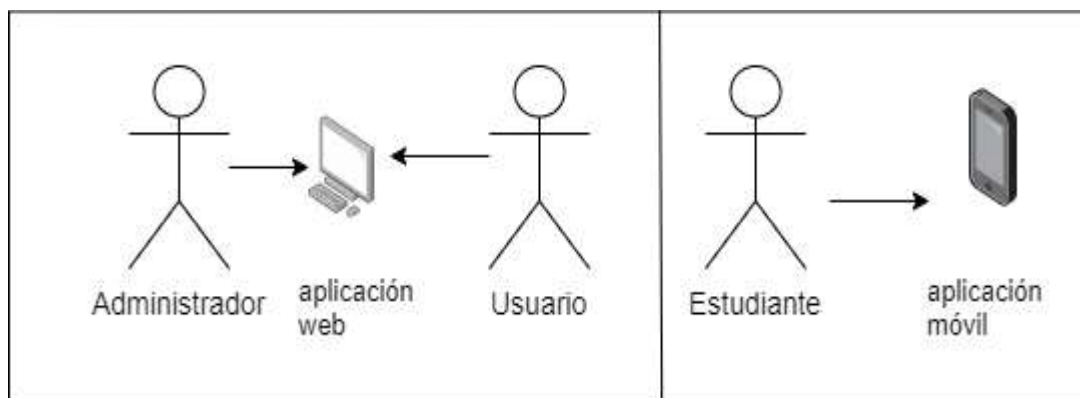


Figura 2.3 Actores que interactúan con el sistema.

La Tabla 2.12 muestra los módulos que conforman el sistema, que a su vez agrupan los diferentes RF que permitirán a los actores anteriormente descritos realizar todas las acciones necesarias para interactuar con el sistema prototipo. Es importante mencionar que estas acciones cumplen con los RF descritos según las necesidades de la AEIE.

Tabla 2.12 Módulos que conforman el prototipo de software distribuido.

MÓDULO	CASO DE USO	RF
Catálogo	CU01	RF01, RF02, RF03, RF04, RF05, RF06, RF07, RF08, RF09.
Ventas	CU02	RF110, RF11.
Servicios	CU03	RF12, RF13, RF14, RF15, RF16, RF17, RF18, RF19, RF20, RF21, RF22, RF23.
Cursos	CU04	RF24, RF25, RF26, RF27.
Informativo	CU05	RF28, RF29, RF30.
Usuarios	CU06	RF31, RF32, RF33, RF34.
Móvil	CU07	RF35, RF36, RF37, RF38.

El módulo **Catálogo** mostrado en la Figura 2.4 se encarga de la gestión de los productos ofertados por la AEIE. Permite añadir, modificar y eliminar la categoría a la que pertenece un producto. Además, añade, modifica o elimina un proveedor de productos. Finalmente, se encarga de añadir, modificar y eliminar un producto. Es importante mencionar que para añadir un producto es necesario seleccionar una categoría y un proveedor para el mismo.

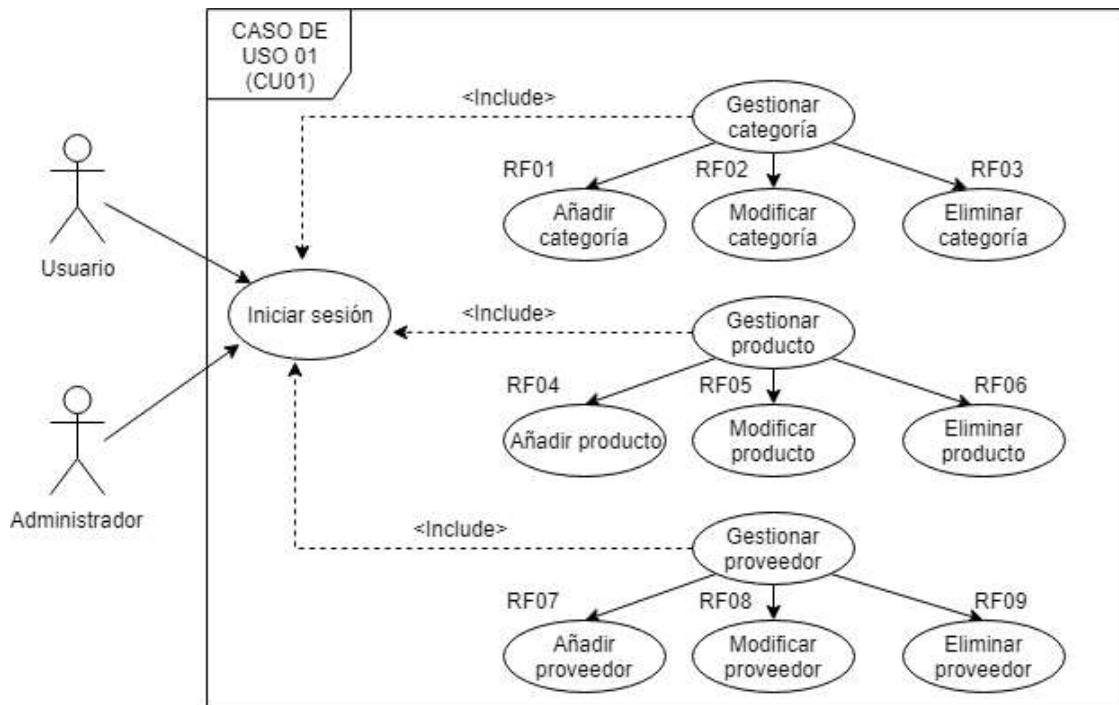


Figura 2.4 Caso de uso del módulo Catálogo.

El módulo **Ventas** indicado en la Figura 2.5 es el encargado de gestionar las ventas de la AEIE. Un Usuario o Administrador podrá registrar una venta. Además, podrá listar todas las ventas realizadas, esta lista indicará quien realizo la venta.

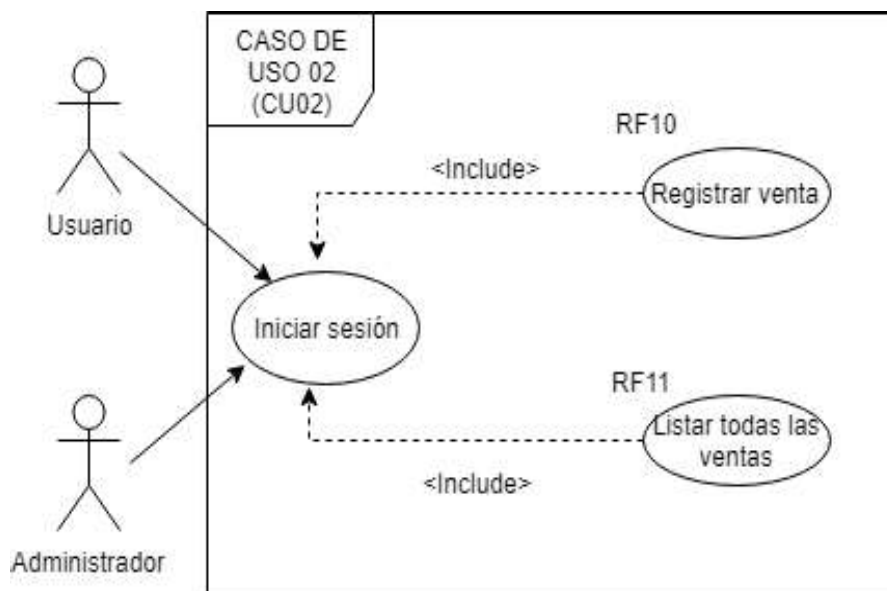


Figura 2.5 Caso de uso del módulo Ventas.

La Figura 2.6 corresponde al módulo **Servicios**; permitirá la gestión de ciertos servicios ofertados por la AEIE. El módulo permite el registro de un semestre referencial. Respecto a los estudiantes permite añadir, modificar, y eliminar estudiantes. Por otra parte, el módulo posibilitará añadir, modificar y eliminar un casillero.

Un Usuario o Administrador que usa el sistema estará en capacidad de registrar, modificar o eliminar el aporte realizado por un estudiante. Así como también, gestionar el alquiler de un casillero por parte de un estudiante, añadiendo o eliminando el registro del mismo. Es importante mencionar que la notificación del estado del casillero se enviará mediante un correo electrónico de forma automática, notificando al estudiante que deberá retirar sus pertenencias cuando se termine el alquiler del casillero en caso de no renovarlo.

El módulo **Cursos** indicado en la Figura 2.7, permitirá la gestión de los cursos ofertados por la AEIE, es decir, registro, modificación y eliminación de un curso. Además, ofrece la capacidad de visualizar la lista de cursos registrados y añadir a un Estudiante en el curso de su preferencia.

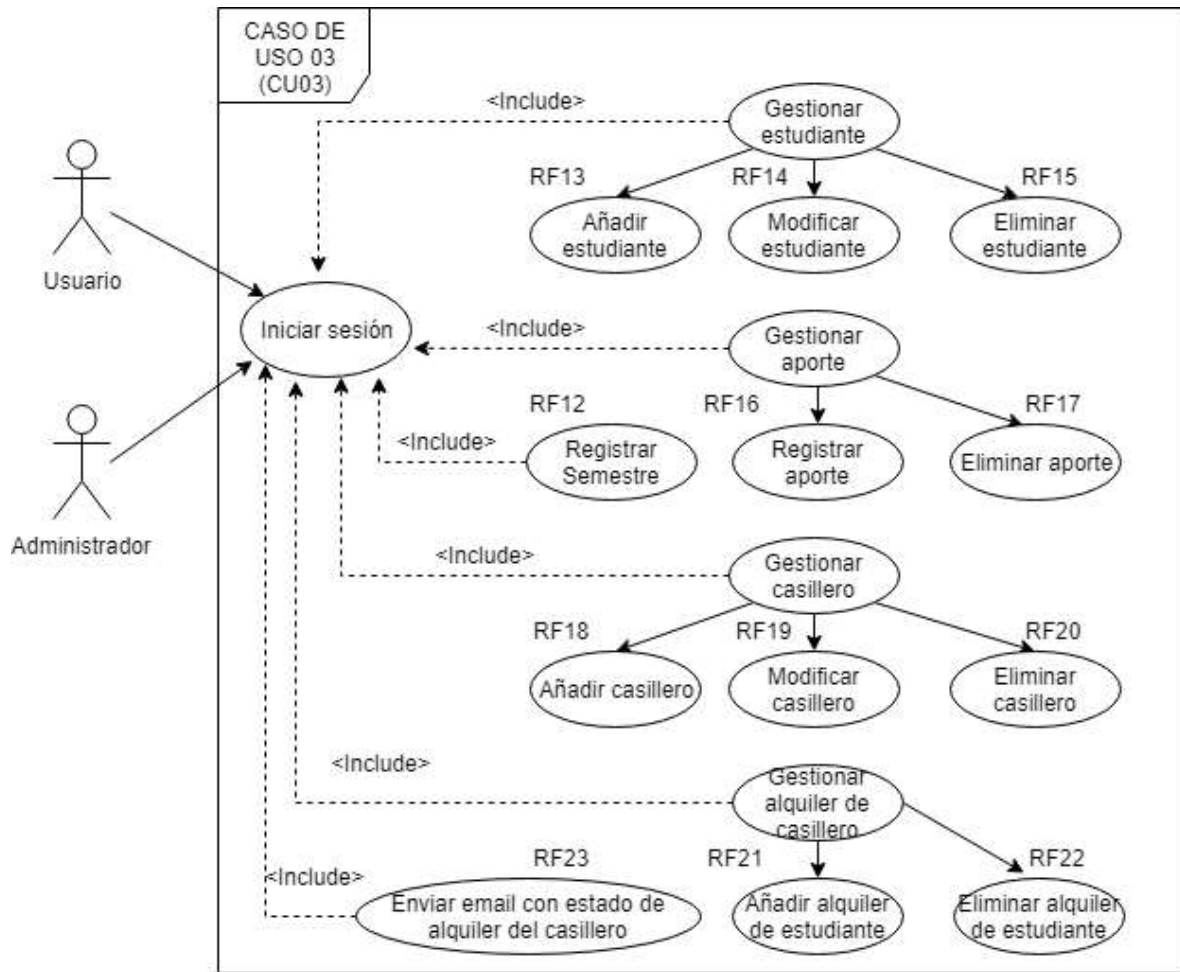


Figura 2.6 Caso de uso del módulo Servicios.

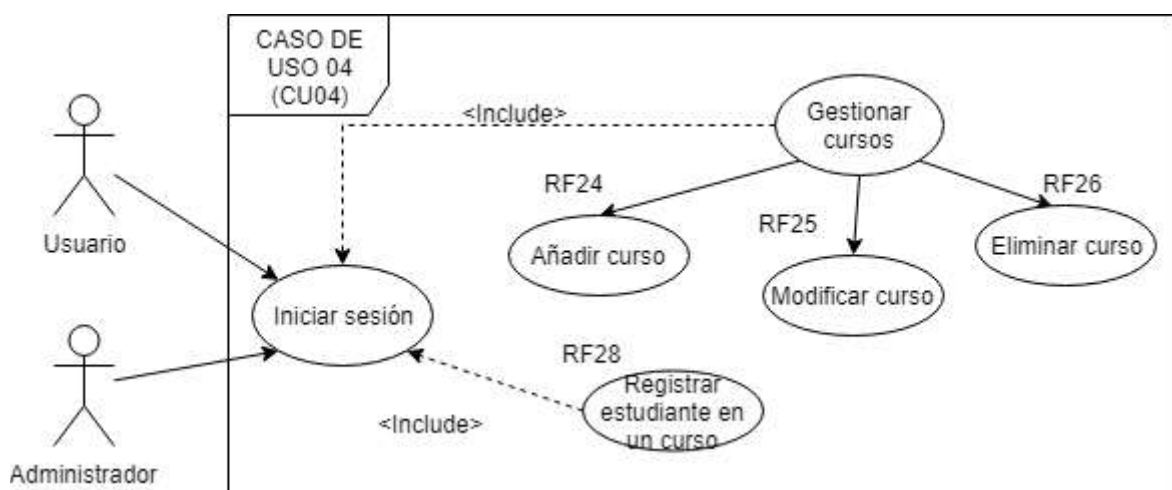


Figura 2.7 Caso de uso del módulo Cursos.

El módulo **Informativo** que se muestra en la Figura 2.8, es el encargado de publicar, modificar y eliminar avisos de tipo informativo por parte de la AEIE. Estos avisos serán visualizados por los Estudiantes en la aplicación móvil, correspondiente al módulo **Móvil**.

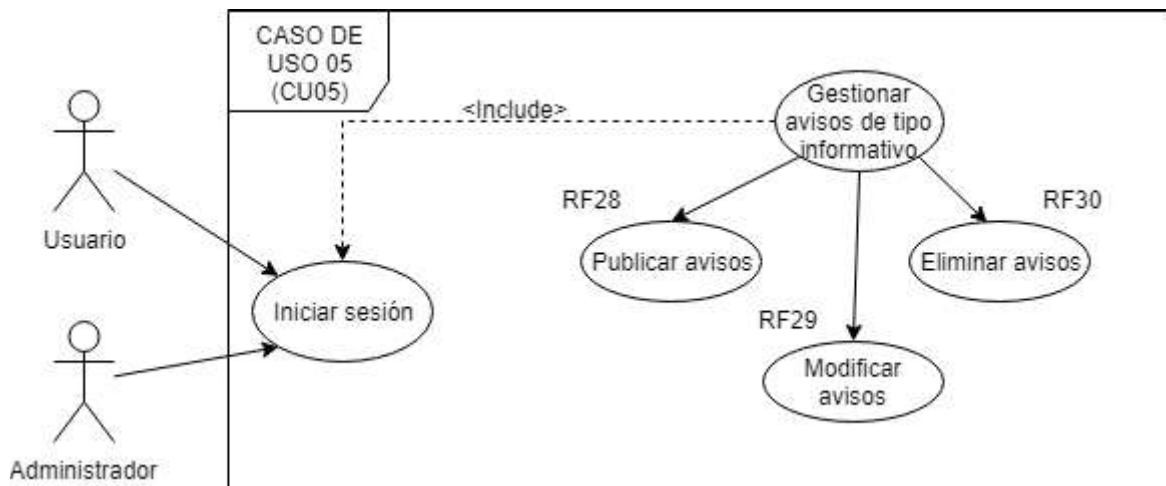


Figura 2.8 Caso de uso del módulo Informativo.

El módulo **Usuarios** presentado en la Figura 2.9, será manipulado y estará accesible únicamente para un Administrador. Este módulo permitirá la creación de usuarios que estén en capacidad de iniciar sesión en el sistema, así como también modificar y eliminar un Usuario.

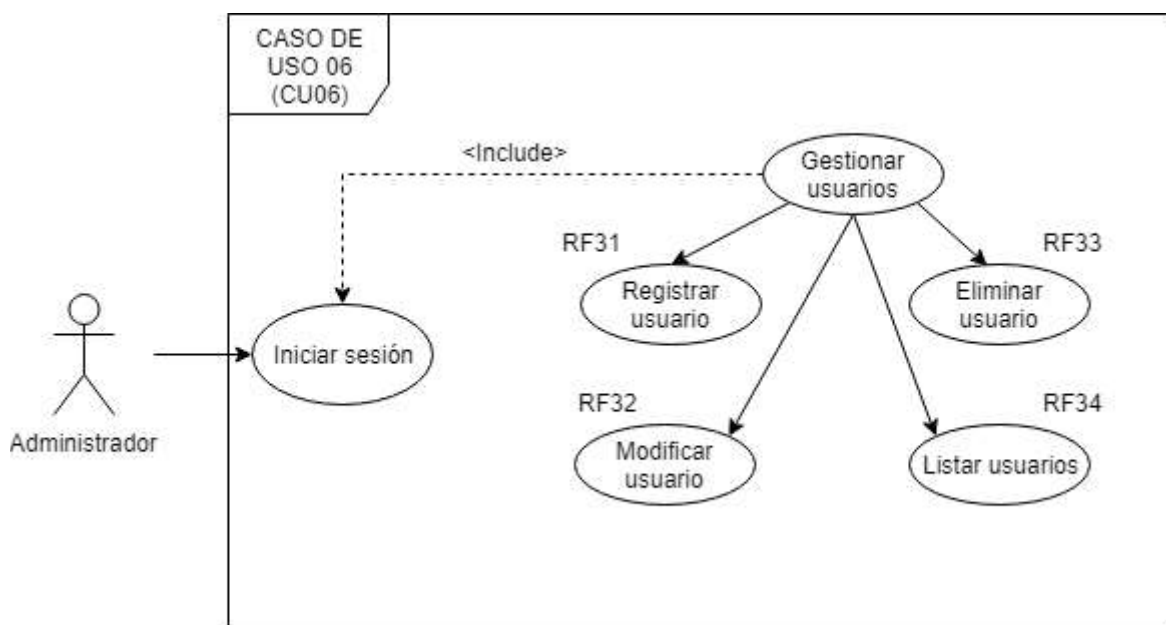


Figura 2.9 Caso de uso del módulo Usuarios.

El módulo **Móvil** corresponde a la aplicación móvil orientada a los Estudiantes y se lo visualiza en la Figura 2.10. El Estudiante podrá visualizar los productos y cursos

ofertados por la AEIE. Además, se podrá observar la disponibilidad de los casilleros en la asociación. Finalmente, el aplicativo móvil permitirá visualizar los avisos de tipo de informativo que son publicados a discreción de la AEIE.

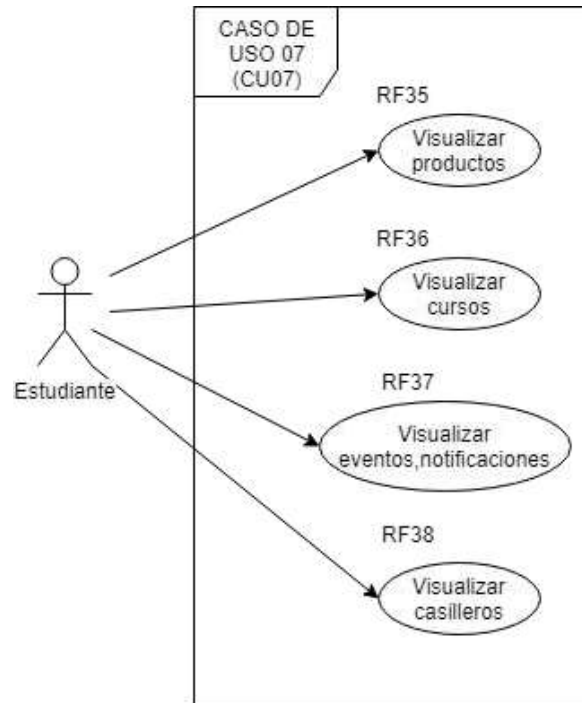


Figura 2.10 Caso de uso del módulo Móvil.

2.2.5.2 Diagrama de Clases

Los Diagramas de Clases son uno de los tipos de diagramas más útiles en (*Unified Modeling Language*) UML⁸, ya que trazan claramente la estructura de un sistema concreto al modelar sus clases, atributos, operaciones y relaciones entre objetos. Los diversos componentes en un diagrama de clases pueden representar las clases que se programarán en realidad, los objetos principales o la interacción entre clases y objetos [38]

. La figura de clase en sí misma consiste en un rectángulo de tres filas. La fila superior contiene el nombre de la clase, la fila del centro contiene los atributos de la clase y la última expresa los métodos o las operaciones que la clase puede utilizar. Las clases y las subclasses se agrupan para mostrar la relación estática entre cada objeto.

En la Figura 2.11 se muestra el diagrama de clases correspondiente al sistema de gestión de productos y servicios ofertados por la AEIE. Este diagrama fue realizado mediante el software de diagramas en línea gratuito, Draw.io [39].

⁸ UML: es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.

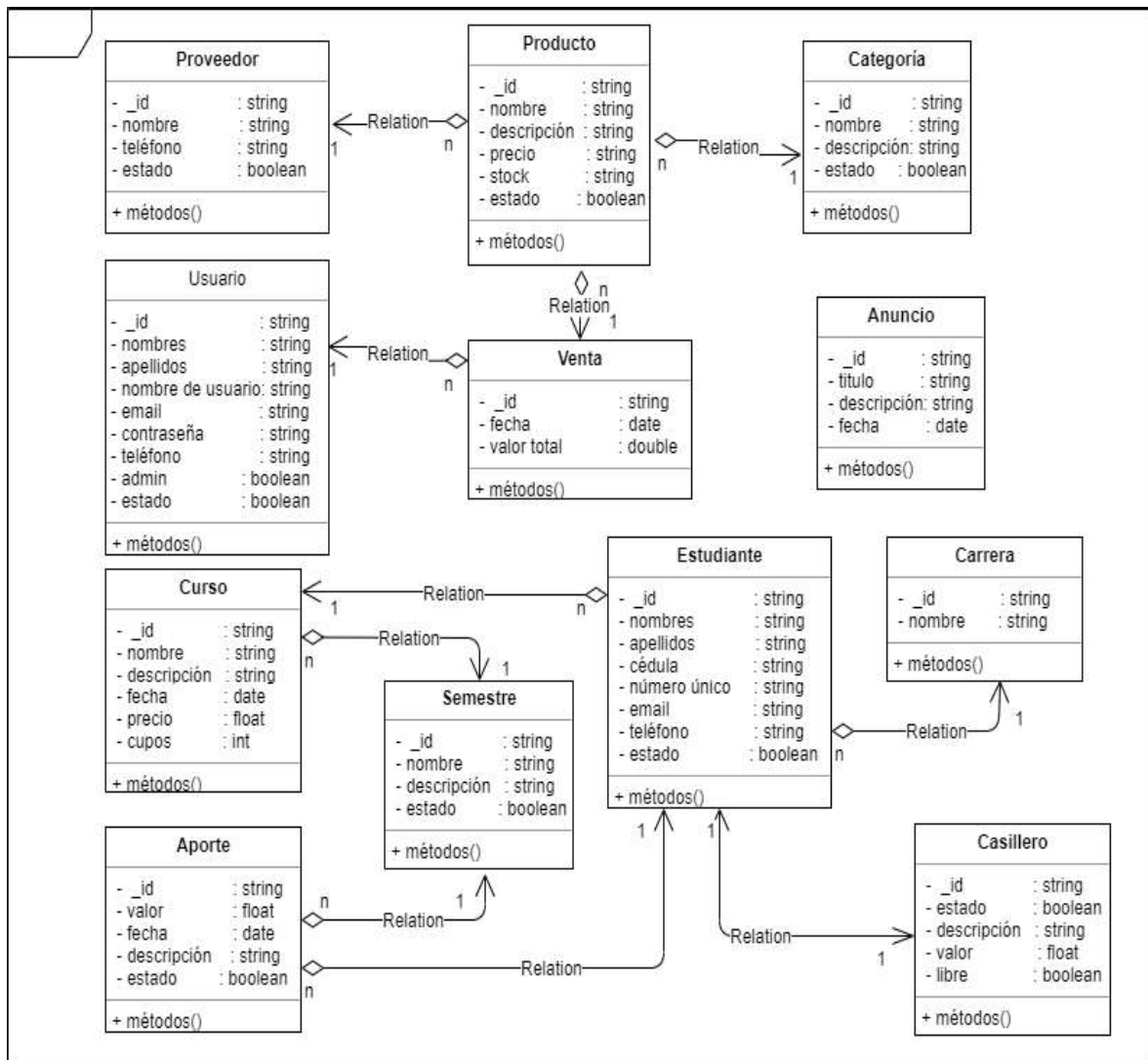


Figura 2.11 Diagrama de clases.

Por falta de espacio no se ha colocado todos los métodos, sin embargo es importante mencionar que los métodos presentes en el Diagrama de Clases de la Figura 2.11, son métodos incorporados (*built-in*) pertenecientes a la librería “Mongoose” y que permiten ejecutar acciones CRUD (*Create, Read, Update, Delete*).

En el diagrama, la clase “*Categoría*” y la clase “*Producto*” tienen la multiplicidad de “uno” a “n” de modo que una categoría puede contener varios productos. La clase “*Proveedor*” y “*Producto*” tienen una multiplicidad de “uno” a “n” demostrando que un proveedor puede abastecer distintos productos. Por otro lado, la clase “*Venta*” posee una multiplicidad de “uno” a “n” con las clases “*Usuario*” y “*Producto*” puesto que un usuario podrá registrar la venta de diversos productos. La clase “*Anuncio*” se encarga de almacenar la información de los avisos de tipo informativo.

Conjuntamente, se tiene la clase “Semestre” que tiene una multiplicidad de “uno” a “n” con las clases “Curso” y “Aporte” encargadas del registro de cursos y aporte de estudiantes respectivamente. La clase “Carrera” muestra una multiplicidad de “uno” a “n” con la clase “Estudiante” indicando que una carrera puede tener muchos estudiantes. Mientras que la multiplicidad “uno” a “uno” entre la clase “Casillero” y “Estudiante” denota la pertenencia que tiene un estudiante al alquilar un casillero.

A modo de ejemplo, se observa el detalle de la clase “Carrera” en la Tabla 2.13. El detalle de las clases restantes se encuentra en el **ANEXO C**.

Tabla 2.13 Detalla de la Clase Carrera.

Carrera		
Atributo	Tipo de dato	Descripción
_id	String	Identificador único de una carrera.
Nombre	String	Nombre referencial para la carrera.

2.2.6 DISEÑO DE LA CAPA DE PRESENTACIÓN

En esta capa se muestran las interfaces que componen el sistema prototipo y que interactuarán con el usuario final. Estas interfaces corresponderán a la aplicación móvil y web. El diseño de los bosquejos para la Capa de Presentación se realizará mediante una herramienta en línea para realizar maquetación y esquema de páginas con el nombre de Ninjamock [40]. Es importante mencionar que las interfaces se han diseñado y organizado mediante módulos para cumplir con las necesidades de la AEIE.

2.2.6.1 Diseño de las Vistas de la Aplicación Web

El sistema web contará con diferentes interfaces gráficas. A continuación, se presenta los distintos bosquejos realizados para los módulos del sistema detallados en el **Apartado 2.2.2**.

La Figura 2.12 muestra el diseño de la página de inicio de sesión, necesaria para utilizar los módulos del sistema. El Usuario ingresa un correo electrónico y contraseña para el inicio de sesión. Es importante mencionar que el Usuario debe ser previamente creado por un Administrador.

La Figura 2.13 muestra un bosquejo de la interfaz para el módulo Catálogo encargado de la gestión de productos. Aquí se ha organizado los RF afines, colocándolos en tres submenús los cuales son: “Producto”, “Categoría” y “Proveedor”. Por ejemplo en el caso

del submenú “Producto”, se tiene dos pestañas que brindan no solo organización sino también una forma amigable e intuitiva para cumplir con los RF pertenecientes a este módulo.



Figura 2.12 Interfaz para el inicio de sesión.



Figura 2.13 Interfaz para el módulo Catálogo.

La Figura 2.14 presenta el bosquejo para el módulo Ventas encargado de la gestión de ventas. Los RF de este módulo se han colocado en un submenú en la parte lateral derecha, y son: “Registrar Venta” y “Todas las ventas”. En la Figura 2.14 se observa el ejemplo cuando el Usuario se encuentra en el submenú “Todas las ventas”. En dicho

submenú el Usuario podrá observar la lista de las ventas realizadas y quién fue el Usuario que las realizó.



Figura 2.14 Interfaz para el módulo Ventas.

La Figura 2.15 muestra el bosquejo correspondiente al módulo Servicios. Se ha organizado los RF establecidos anteriormente en la Tabla 2.11 mediante submenús. Se puede observar un ejemplo correspondiente al registro de Estudiantes en el sistema. La interfaz mostrará la lista de Estudiantes registrados y a su derecha las acciones como modificar o eliminar un Estudiante.

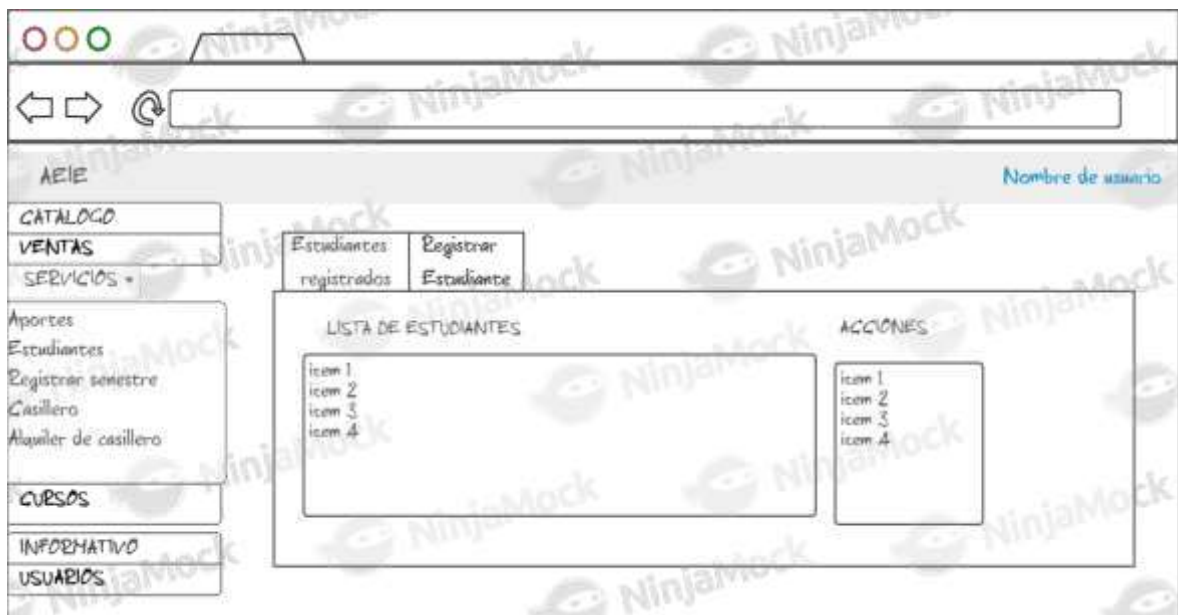


Figura 2.15 Interfaz para el módulo Servicios.

En la Figura 2.16 se visualiza el bosquejo de la interfaz perteneciente al módulo Cursos. Al expandir el menú se tiene los siguientes submenús: “Registrar curso”, “Registrar estudiantes”.

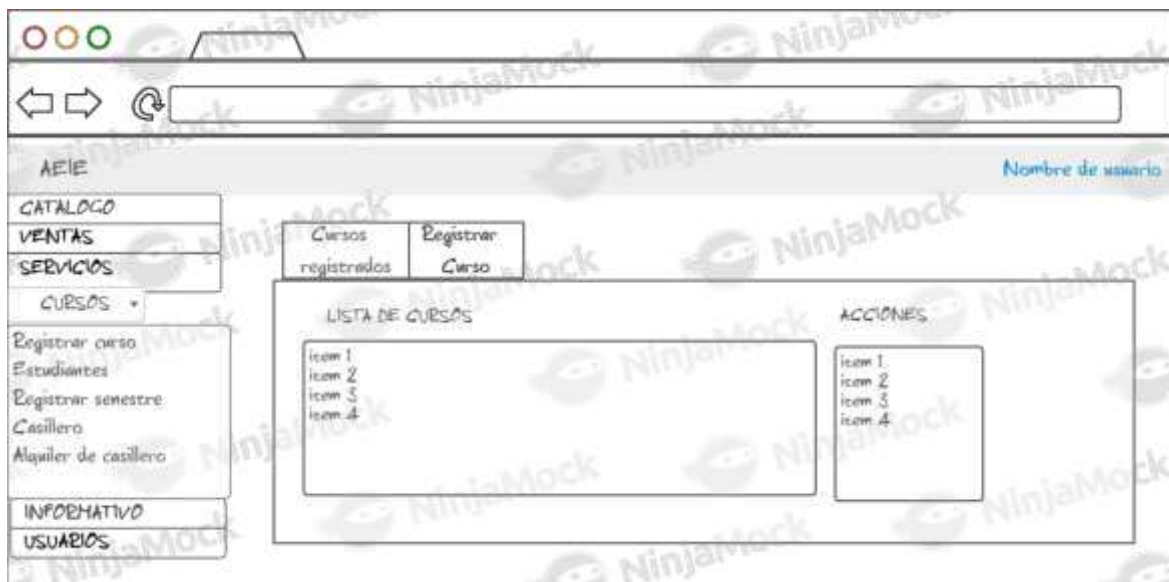


Figura 2.16 Interfaz para el módulo Cursos.

La Figura 2.17 muestra el bosquejo de la interfaz para el módulo Informativo. Este módulo se encarga de publicar anuncios de tipo informativo. Por ejemplo, la pestaña “Registrar Anuncio” servirá para publicar los anuncios a discreción de la AEIE, los cuales tendrán un título y una descripción.

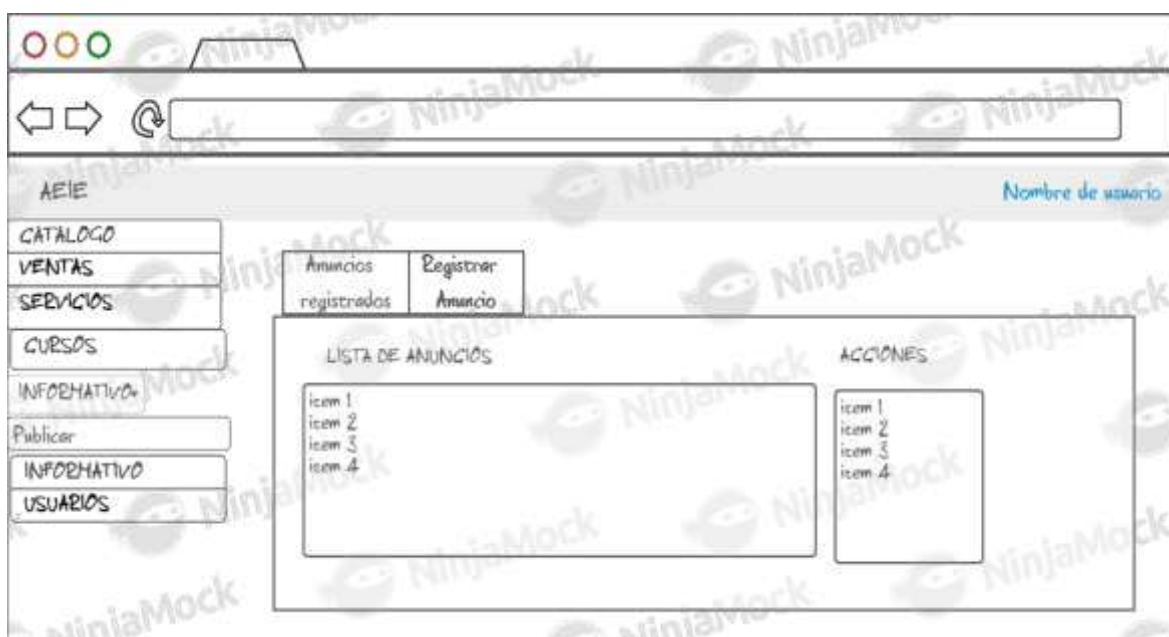


Figura 2.17 Interfaz para el módulo Informativo.

El bosquejo de la interfaz del módulo Usuarios se observa en la Figura 2.18. Este es el único módulo que será manipulado por un Administrador. Un Usuario será de tipo Administrador cuando se configure esta propiedad al momento de su creación en la pestaña “Registrar Usuario”. Este módulo permitirá la creación de Usuarios que estén en capacidad de iniciar sesión en el sistema, así como también modificar y eliminar un Usuario.

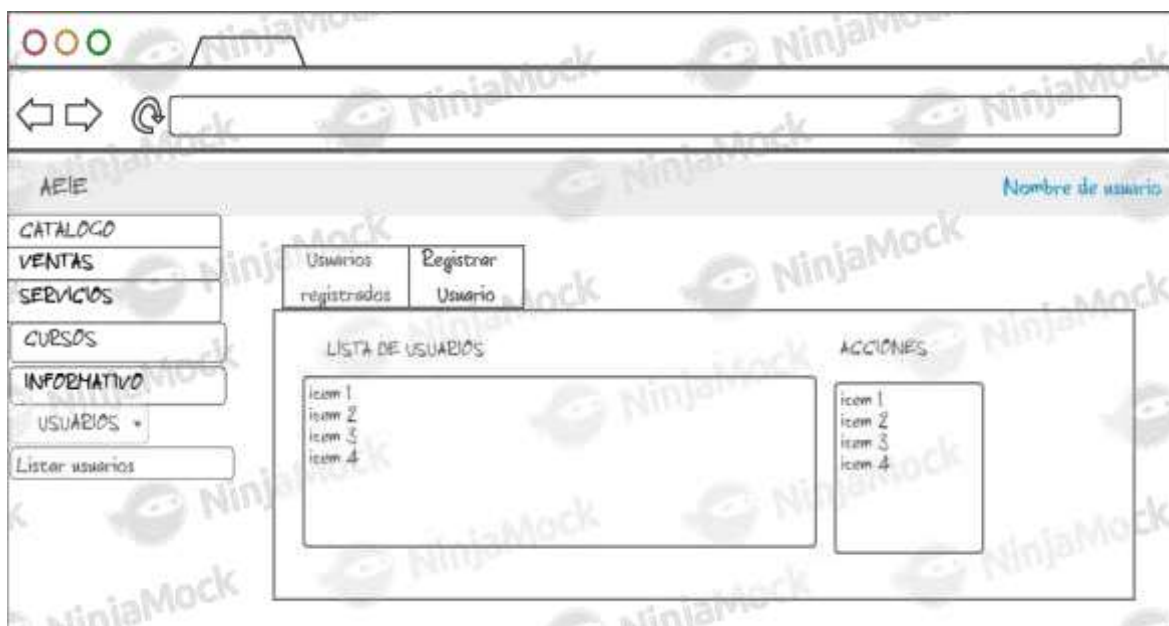


Figura 2.18 Interfaz para el módulo Usuarios.

2.2.6.2 Diseño de las Vistas de la Aplicación Móvil.

Para el módulo Móvil correspondiente a la aplicación móvil se organizó los RF establecidos en la Tabla 2.11 en cuatro menús que son:

- Productos.
- Cursos.
- Casilleros.
- Anuncios.

En la Figura 2.19 se observa los bosquejos correspondientes a los menús “*Productos*” y “*Casilleros*”. En el primero se tiene una lista de los productos y en el segundo los cursos ofertados por la AEIE. Así también, en la Figura 2.20 se presenta los bosquejos de los menús “*Casilleros*” y “*Anuncios*”, visualizando la disponibilidad de los casilleros y los anuncios de tipo informativo.



Figura 2.19 Interfaz módulo Móvil de los menús “Productos” y “Casilleros”.



Figura 2.20 Interfaz módulo Móvil de los menús “Cursos” y “Anuncios”.

2.3 IMPLEMENTACIÓN

En esta etapa, primeramente se realizará la actualización del Tablero Kanban con las tareas necesarias para cumplir con la etapa de implementación. Se establecerá el ambiente adecuado para la codificación, el cual comprende los entornos de desarrollo integrado (IDE) que se ajusten a la necesidad del presente proyecto. Además, se instalarán las herramientas necesarias para el desarrollo del prototipo. Se codificará la Capa de Datos, Negocio y Presentación correspondiente al prototipo diseño. Finalmente, se codificará la aplicación móvil Android, basándose en los requerimientos definidos en el **Apartado 2.2.2.2**.

2.3.1 ACTUALIZACIÓN DEL TABLERO KANBAN

El tablero con las tareas actualizadas se presenta en la Figura 2.21, este muestra el avance del Trabajo de Titulación. Una vez cumplidas con las tareas de la etapa de Diseño, estas son dirigidas a la columna del extremo de derecho correspondiente a las “TAREAS REALIZADAS”. Mientras que en la columna central perteneciente a las “TAREAS EN PROCESO”, se han colocado las tareas correspondientes a la fase de Implementación. Por último, la columna “LISTA DE TAREAS” muestra las tareas que serán llevadas a cabo en la fase de pruebas.

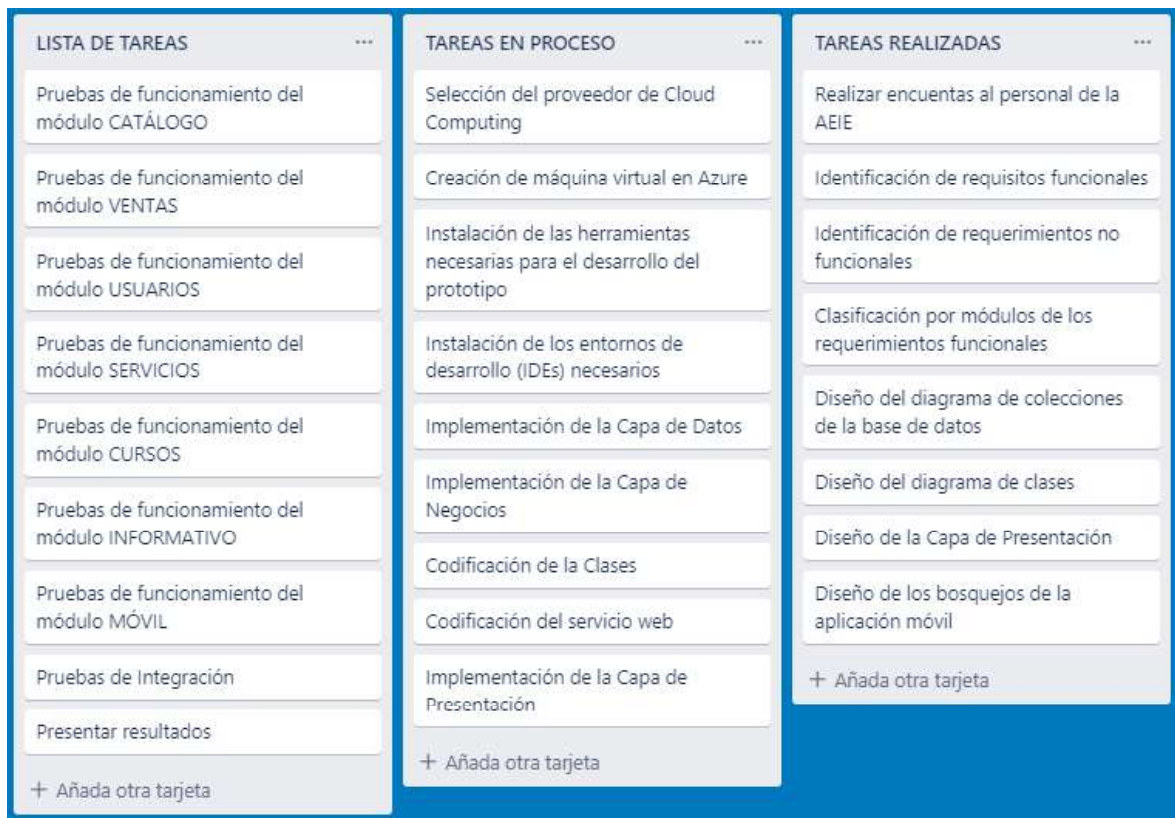


Figura 2.21 Tablero Kanban para la fase de implementación.

2.3.2 INSTALACIÓN DE LAS HERRAMIENTAS NECESARIAS

En este apartado se expone la instalación de las herramientas necesarias que permitirán la implementación del prototipo de software distribuido. Así como también, se acondicionará el servidor donde se alojarán las capas del prototipo de software distribuido.

2.3.2.1 Creación de una máquina virtual en Microsoft Azure.

Para el acondicionamiento del servidor, se creará una máquina virtual desde el portal de Microsoft Azure. Una vez iniciada la sesión en Microsoft Azure, el proceso consiste en seleccionar la opción “*Crear un recurso*” ubicada dentro del panel principal en la esquina superior izquierda como se observa en la Figura 2.22.



Figura 2.22 Creación de un recurso a través del Portal Microsoft Azure.

Como siguiente paso, entre los distintos tipos de servicios ofertados por Microsoft Azure se escoge “*Proceso*” y a continuación “*Máquina Virtual*”. Estos pasos están indicados en la Figura 2.23.

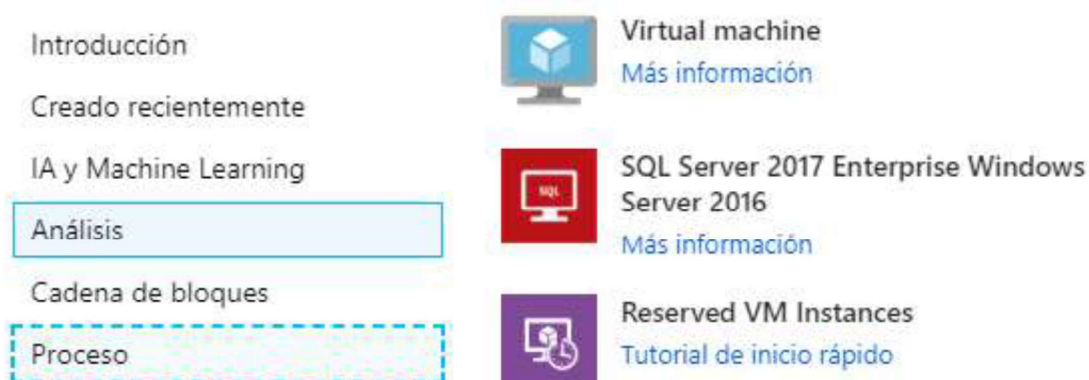


Figura 2.23 Creación de una máquina virtual en Microsoft Azure.

La Figura 2.24 muestra la pestaña “Aspectos básicos” correspondiente a la máquina virtual que se está creando. Aquí se seleccionará el “Grupo de recursos⁹” donde se alojará la máquina virtual. En este caso se le ha asignado el nombre de “tesisaeie”.

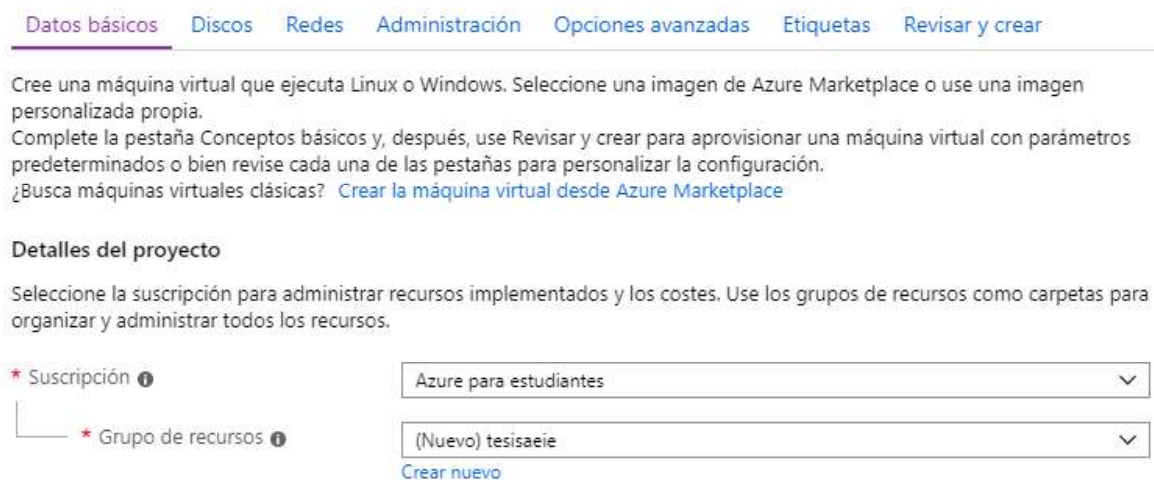


Figura 2.24 Conceptos básicos de una máquina virtual en Azure.

La Figura 2.25 presenta la sección “Detalles de Instancia”. Aquí, se colocará un nombre de identifique a nuestra máquina virtual, por ejemplo “tesisaeieserver”. Además, se seleccionará el sistema operativo que en este caso será “Ubuntu Server 18.04 LTS”. En cuanto al tamaño, la máquina virtual corresponde al tipo “B1S” con una unidad CPU virtual y memoria RAM de 1GB.



Figura 2.25 Detalles de instancia de una máquina virtual en Azure.

En la sección “Cuenta de administrador”, se selecciona el usuario para la máquina virtual y se añade la clave pública para la conexión remota mediante SSH. En la Figura 2.26 se visualiza el usuario administrador a utilizarse con el nombre de “tesisaeie”.

⁹ *Grupo de recursos*: es una colección de recursos que comparten los mismos permisos, ciclo de vida y directivas.

Cuenta de administrador

Tipo de autenticación ? Contraseña Clave pública SSH

* Nombre de usuario ? ✓

* Clave pública SSH ? ✓

[Más información sobre la creación y el uso de claves SSH en Azure](#)

Figura 2.26 Configuración del acceso remoto a la máquina virtual Azure.

Para el acceso a la máquina virtual a través de internet, se habilitará los puertos principales que son: HTTP (80), HTTPS (443) y SSH (22); como se observa en la Figura 2.27. El primero y segundo puertos permitirán el acceso a la aplicación web, mientras que el tercero posibilitará el acceso remoto para la administración del servidor.

Reglas de puerto de entrada

Seleccione los puertos de red de máquina virtual que son accesibles desde la red Internet pública. Puede especificar acceso de red más limitado o granular en la pestaña Red.

* Puertos de entrada públicos ? Ninguno Permitir los puertos seleccionados

* Seleccionar puertos de entrada ✓

! Estos puertos se exponen a Internet. Use los controles avanzados para limitar el tráfico entrante a las direcciones IP conocidas. También puede actualizar las reglas de tráfico entrante más adelante.

Figura 2.27 Reglas de puerto de entrada para la máquina virtual.

Una vez que los pasos anteriores se han llevado a cabo y se revisa los detalles de la máquina virtual, solo resta dar *click* en el botón “*Revisar y Crear*”. Finalmente, cuando la máquina virtual esté lista se comprobará la conexión remota mediante *PuTTY*¹⁰ como se visualiza en la Figura 2.28.

```
Last login: Tue Sep 17 18:06:37 2019
tesisaeie@tesisaeieserver:~$
```

Figura 2.28 Prueba de conexión remota mediante SSH.

2.3.2.2 Instalación MongoDB.

Primero, se importa la clave pública usada por el sistema que asegura la consistencia y autenticidad del paquete a instalarse. Segundo, se añade el repositorio de MongoDB al sistema. Tercero, se actualiza la base de datos de paquetes local. Cuarto, se instala

¹⁰ PuTTY: es un cliente SSH y telnet.

MongoDB junto a todos sus paquetes. Finalmente, se inicia el servicio en Ubuntu para poder utilizar MongoDB.

```
1 wget -qO - https://www.mongodb.org/static/pgp/server-4.2.asc | sudo
  apt-key add -
2 echo "deb [ arch=amd64 ] https://repo.mongodb.org/apt/ubuntu bionic/
  mongodb-org/4.2 multiverse" | sudo tee /etc/apt/sources.list.d/
  mongodb-org-4.2.list
3 sudo apt-get update
4 sudo apt-get install -y mongodb-org
5 sudo service mongod start
```

Código 2.1 Comandos para instalar MongoDB.

2.3.2.3 Instalación Node.js y Express.

Primero, hay que instalar el PPA (*Personal Package Active*) para obtener acceso a su contenido. Mediante el comando *curl* se obtiene el script de instalación para la versión de Node.js preferida; en este caso será la versión 10.x. Una vez que el PPA se agrega a la configuración y la caché del paquete local en Ubuntu, se actualizará automáticamente. Después de ejecutar el script de configuración descargado desde Nodsource, se puede instalar el paquete Node.js ejecutando el comando “*apt-get install*”.

```
1 curl -sL https://deb.nodesource.com/setup_10.x | sudo -E bash -
2 sudo apt-get install -y nodejs
```

Código 2.2 Comandos para instalar Node.js.

El paquete de Node.js instalado en Ubuntu ya contiene npm¹¹, así que no es necesario instalarlo por separado. Es así, que para instalar Express basta con ejecutar el comando mostrado en el Código 2.3.

```
1 npm install express --save
```

Código 2.3 Comando para instalar Express.

2.3.2.4 Instalación Angular.

Para su instalación simplemente se abre un terminal y se ejecuta el comando indicado en el Código 2.4. Angular requiere que Node.js esté instalado previamente.

```
1 npm install -g @angular/cli
```

Código 2.4 Comando para instalar Angular.

¹¹ npm: es un administrador de paquetes para Node.js.

2.3.2.5 Instalación Visual Studio Code.

Desde la página oficial [41] de Visual Studio Code se puede descargar el instalador de acuerdo al sistema operativo de preferencia. Una vez descargado el instalador, se ejecuta para poner en marcha el asistente de instalación y esto tomará unos minutos. Cuando se ha instalado por completo el software, se muestra la página de bienvenida como se visualiza en la Figura 2.29.

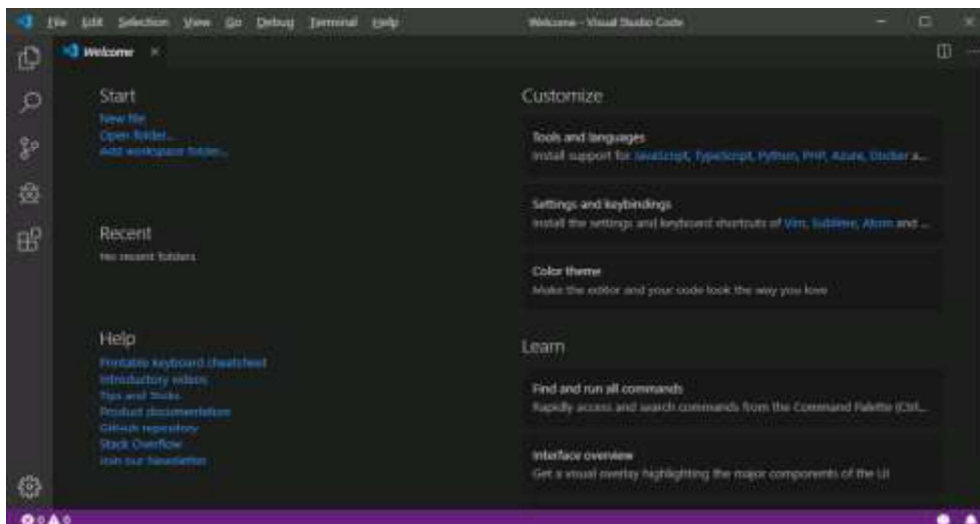


Figura 2.29 Pantalla de bienvenida en Visual Studio Code.

2.3.2.6 Instalación Android Studio

Primero, se realiza la descarga de la versión reciente del ejecutable de Android Studio desde la página oficial [42]. Una vez descargado el archivo ".exe", hay que hacer doble *click* para ejecutarlo. A continuación, se sigue la guía del asistente de configuración de Android Studio. Finalmente, terminada la instalación aparece la pantalla de bienvenida mostrada en la Figura 2.30.



Figura 2.30 Pantalla de Bienvenida de Android Studio.

2.3.3 IMPLEMENTACIÓN DE LA CAPA DE DATOS

En esta apartado, se presentará un ejemplo para la codificación de la Capa de Datos basado en el **Apartado 2.2.4** correspondiente a la etapa de Diseño. Con el propósito de aplicar buenas prácticas de programación en la etapa de implementación, se ha codificado en idioma Inglés las Capas de Datos, Lógica de Negocio y Presentación. Las vocales acentuadas y la ñ, al no ser caracteres básicos ASCII, pueden sufrir alteraciones cuando un fichero de código se abre en dos computadoras diferentes, incluso si ambas contienen el mismo sistema operativo. Para evitar estos problemas es recomendable codificar directamente en inglés [43].

El Código 2.5 corresponde a la creación de la colección “*Product*” en la base de datos. Esta tarea se realizará mediante “*Mongoose*”, la cual es una librería que mapea un objeto a documento, la cual aprovecha la versatilidad de la Programación Orientada a Objetos abstrayendo al programador de la base de datos y centrándolo en el desarrollo de la aplicación. Otro punto a favor, es la facilidad de trabajo. “*Mongoose*” facilita las labores de acceso a los datos como: crear, leer, actualizar y eliminar.

En la línea 1 del Código 2.5 se importa la librería “*Mongoose*”. Dado que en “*Mongoose*” todo empieza con un esquema (*Schema*), se define el esquema para la colección “*Product*” entre las líneas 3 al 15. El esquema se asigna a una colección en MongoDB y define la estructura que tendrán los *documentos* en la colección. Cada clave de “*productSchema*” define la propiedad y el tipo de dato que tendrá el *documento*. Por ejemplo las propiedades definidas en el Código 2.5 corresponden a:

- “*_id*”: Identificador único para un producto.
- “*name*”: Nombre del producto.
- “*description*”: Información adicional que caracterice a un producto.
- “*price*”: Precio unitario de un producto.
- “*stock*”: Cantidad de productos disponibles.
- “*category*”: Identificador (*_id*) correspondiente a la categoría de un producto.
- “*supplier*”: Identificador (*_id*) correspondiente al proveedor de un producto.
- “*state*”: Indicador para ocultar el producto cuando se elimine.

Para poder usar el esquema definido, es necesario convertir el “*productSchema*” en un modelo con el que se pueda trabajar. Los modelos son los responsable de crear y leer

documentos en la base de datos de MongoDB, esta acción se encuentra reflejada en la línea 18. El código restante correspondiente a la implementación de las demás colecciones se encuentra en el **ANEXO D**.

```
1  const mongoose = require("mongoose");
2
3  const productSchema = mongoose.Schema({
4    _id: mongoose.Schema.Types.ObjectId,
5    name: { type: String, required: true },
6    description: { type: String },
7    price: { type: Number, required: true },
8    stock: { type: Number, required: true },
9    category: {
10     type: mongoose.Schema.Types.ObjectId,
11     ref: "Category",
12     required: true
13   },
14   supplier: { type: mongoose.Schema.Types.ObjectId, ref: "Supplier" },
15   state: { type: Boolean, default: true }
16 });
17
18 module.exports = mongoose.model("Product", productSchema);
```

Código 2.5 Código para la creación de la colección "Producto".

2.3.4 IMPLEMENTACIÓN DE LA CAPA DE LÓGICA DE NEGOCIO

A continuación, se presenta la implementación de las clases y del servicio web.

2.3.4.1 Codificación de Clases

En el **Apartado 2.3.3**, se ha explicado que la Capa de Datos hará uso de Mongoose, permitiendo el acceso desde la Capa de Negocio hacia la de Capa de Datos sin necesidad de manipular la base de datos manualmente. Para esto se definen Modelos a través de Mongoose encargados de manipular los documentos en MongoDB. Estos Modelos se encargarán de gestionar el acceso a las colecciones, apoyados de métodos incorporados de la librería Mongoose. Entre los principales métodos están: *find()*, *save()*, *updateOne()*, *deleteOne()*, entre otros.

2.3.4.2 Codificación del Servicio Web

En este apartado se utilizó Node.js y Express para la codificación de un servicio web de tipo REST. La Figura 2.31 muestra la carpeta del servicio web con archivos y directorios importantes para su implementación. A continuación, se describen algunos componentes significativos que permiten la funcionalidad del servicio web:

- *Models*: Este directorio contiene los modelos que permitirán ejecutar distintas acciones sobre los documentos en la base de datos.
- *Routes*: Este directorio contiene archivos que manejan las rutas para acceder a los recursos que implementa el servicio web.
- *Controllers*: Este directorio contiene los controladores que procesan las solicitudes entrantes hacia el servicio web.

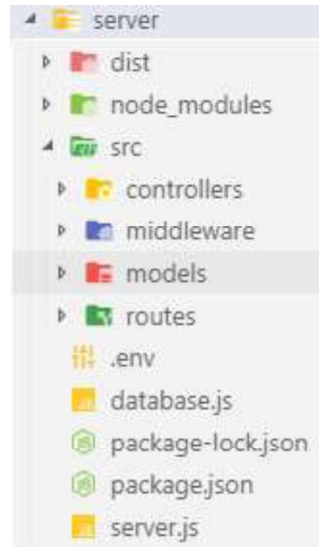


Figura 2.31 Carpeta del servicio web.

El Código 2.6 muestra la implementación de la ruta “*Product*”. Una ruta es una sección de código Express que asocia un verbo HTTP (GET, POST, PUT, DELETE, etc.), a una ruta o patrón de URL, asignando una función que responde a dicha ruta de acceso. En la línea 1 se importa “*express*” a fin de utilizar el objeto “*Router*”, el cual permite definir rutas. Se ha implementado varias rutas, cada una con una acción a ejecutarse. La Tabla 2.14 muestra las acciones que se ejecutarán asociadas a ciertos métodos HTTP en la ruta “*Product*”.

Tabla 2.14 Métodos HTTP asociados a las acciones en la ruta “*Product*”.

Método HTTP	Acción a ejecutarse
POST	Crear un nuevo producto y almacenarlo en la base de datos.
GET	Obtener todos los productos registrados en la base de datos.
PUT	Actualizar un producto almacenado en la base de datos.
DELETE	Eliminar un producto almacenado en la base de datos.

Con el objetivo de entender la implementación, se explica una porción del Código 2.6. En la línea 10 se implementa una ruta que permite crear un nuevo producto y almacenarlo en la base de datos; esta ruta es definida usando el método *Router.post()*, que responde sólo a solicitudes de tipo HTTP POST. El primer argumento de este método es la ruta URL mientras que el segundo es una función que será invocada si una solicitud de tipo HTTP POST con la ruta especificada es recibida.

```
1  const express = require("express");
2  const router = express.Router();
3  const auth = require("../middleware/auth");
4  const ProductsController = require("../controllers/product.controller");
5
6  //Show all products
7  router.get("/", ProductsController.get_all_products);
8
9  //Add a new product
10 router.post("/", auth, ProductsController.create_product);
11 //Show a product by Id
12 router.get("/:productId", ProductsController.get_product);
13 //Update a product by Id
14 router.put("/:productId", auth, ProductsController.update_product);
15 //Delete a product by Id
16 router.delete("/:productId", auth, ProductsController.delete_product);
17
18 module.exports = router;
```

Código 2.6 Archivo que maneja la ruta al recurso Producto.

La función asociada al método POST pertenece al controlador "*ProductsController*". En la línea 20 del Código 2.7, se implementa la función "*create_product*" encargada de crear un nuevo producto y almacenarlo en la base de datos. En la línea 21 se utiliza el modelo "*Product*" definido previamente con Mongoose, este permite guardar el nuevo producto en la base de datos a través del método *save()*.

Las propiedades para el modelo "*Product*" son recibidas mediante el objeto "*req*" (*request*). Este objeto representa a la solicitud HTTP POST, donde las propiedades como: *id*, *name*, *description*, *price*, *stock*, *supplier*, y *category*; vienen en el cuerpo (*body*) de la solicitud. Una vez recibidas las propiedades son asignadas al modelo y esto se visualiza entre las líneas 22 y 28.

Al aplicar el método *save()*, en la línea 21, para guardar los datos en la base; se puede obtener dos resultados. El primer resultado ubicado en la línea 33 del Código 2.7, indica que el proceso fue exitoso y se devuelve el resultado de la solicitud POST mediante el

objeto “res” (*response*). Este objeto responde con el código HTTP (201) explicando que la petición ha sido completada y ha resultado en la creación de un nuevo recurso [44]. Además, envía un mensaje informando que el producto se ha creado exitosamente. Por otra parte, el segundo resultado presente en la línea 41 cumple la función de indicar que se ha producido un error. El error es enviado junto con el código HTTP (500), este código se utiliza cuando se encuentran con situaciones de error ajenas a la naturaleza del servidor web [44]. El código restante del servicio web se encuentra en el **ANEXO D**.

```
20 exports.create_product = (req, res, next) => {
21   const product = new Product({
22     _id: new mongoose.Types.ObjectId(),
23     name: req.body.name,
24     description: req.body.description,
25     price: req.body.price,
26     stock: req.body.stock,
27     category: req.body.category,
28     supplier: req.body.supplier
29   });
30   product
31     .save()
32     .then(result => {
33       res.status(201).json({
34         message: "Created product succesfully",
35         createdProduct: {
36           result
37         }
38       });
39     })
40     .catch(err => {
41       res.status(500).json({
42         error: err
43       });
44     });
45   };
```

Código 2.7 Función "create_product" del controlador "ProductController".

2.3.5 IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN

En este apartado se presenta la implementación de las interfaces gráficas de la aplicación web y móvil según lo expuesto en el **Apartado 2.2.6**.

2.3.5.1 Codificación de la Aplicación Web

Para la ejecución de esta tarea se ha utilizado Angular, esta herramienta fue descrita en el **Apartado 1.3.5.1**. Una aplicación de Angular está compuesta de varios componentes, donde cada componente está a cargo de una responsabilidad específica. En Figura 2.32, se puede visualizar la implementación del Módulo “Catálogo”.

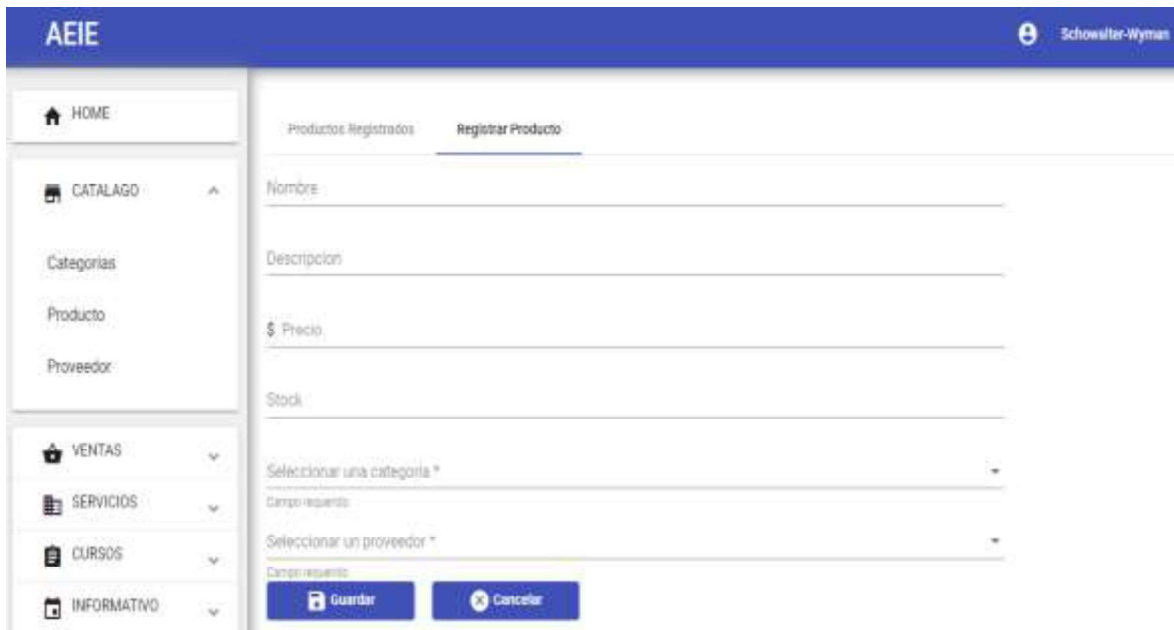


Figura 2.32 Implementación de la interfaz gráfica del Módulo Catálogo.

El módulo “Catálogo” está compuesto de tres componentes, uno para cada submenú mostrado en la Figura 2.33. Es así que se tendrá tres componentes respectivamente y son: “*CategoryComponent*”, “*ProductComponent*”, “*SupplierComponent*”.



Figura 2.33 Menú del Módulo Catálogo.

A continuación, se describe a manera de ejemplo una parte del código perteneciente al módulo “Catálogo”, ya que el código completo estará en el **ANEXO E**. Además, en este mismo anexo se encontrará el código completo del *frontend*¹² de la aplicación web desarrollada con Angular.

¹² *frontend*: describe la parte de una aplicación web que el usuario ve y con la cual interactúa.

Un componente en Angular, en este caso el “*ProductComponent*”, consiste de tres cosas:

- Un *componente de clase* que maneja los datos y la funcionalidad.
- Una *plantilla HTML* que determina la interfaz gráfica.
- *Estilos específicos del componente* que definen como luce, haciendo referencia a los archivos de tipo CCS.

En el Código 2.8 se puede observar la *plantilla HTML* asociada al “*ProductComponent*”. La línea 97 indica la pestaña que contendrá al formulario donde se registrará un nuevo producto. En la línea 98, se define el formulario (form) que contendrá los elementos que permitirán al usuario ingresar las propiedades de un producto. Este formulario tiene asociado el método *saveProduct()* encargado de enviar los datos al servicio web. En línea 111, se visualiza una etiqueta HTML de tipo input en la cual el usuario podrá ingresar el nombre de un producto. Además la línea 120, es utilizada para indicar de forma visual al usuario que el campo nombre es obligatorio.

```
97 <mat-tab label="Registrar Producto">
98   <form
99     #productForm="ngForm"
100     autocomplete="off"
101     (ngSubmit)="saveProduct(productForm)"
102   >
103     <input
104       type="hidden"
105       name="_id"
106       #_id="ngModel"
107       [(ngModel)]=product._id"
108     />
109     <div class="example-container">
110       <mat-form-field [hideRequiredMarker]="true">
111         <input
112           matInput
113           placeholder="Nombre"
114           name="name"
115           [(ngModel)]=product.name"
116           #name="ngModel"
117           required
118           type="text"
119         />
120         <mat-error [hidden]="name.valid || name.pristine"
121           >Campo requerido</mat-error
122         >
123       </mat-form-field>
124     </div>
125   </form>
126 </mat-tab>
```

Código 2.8 Plantilla HTML para el "*ProductComponent*".

El resultado de la definición de los elementos en la *plantilla HTML* del “*ProductComponent*” da como resultado la Figura 2.34.

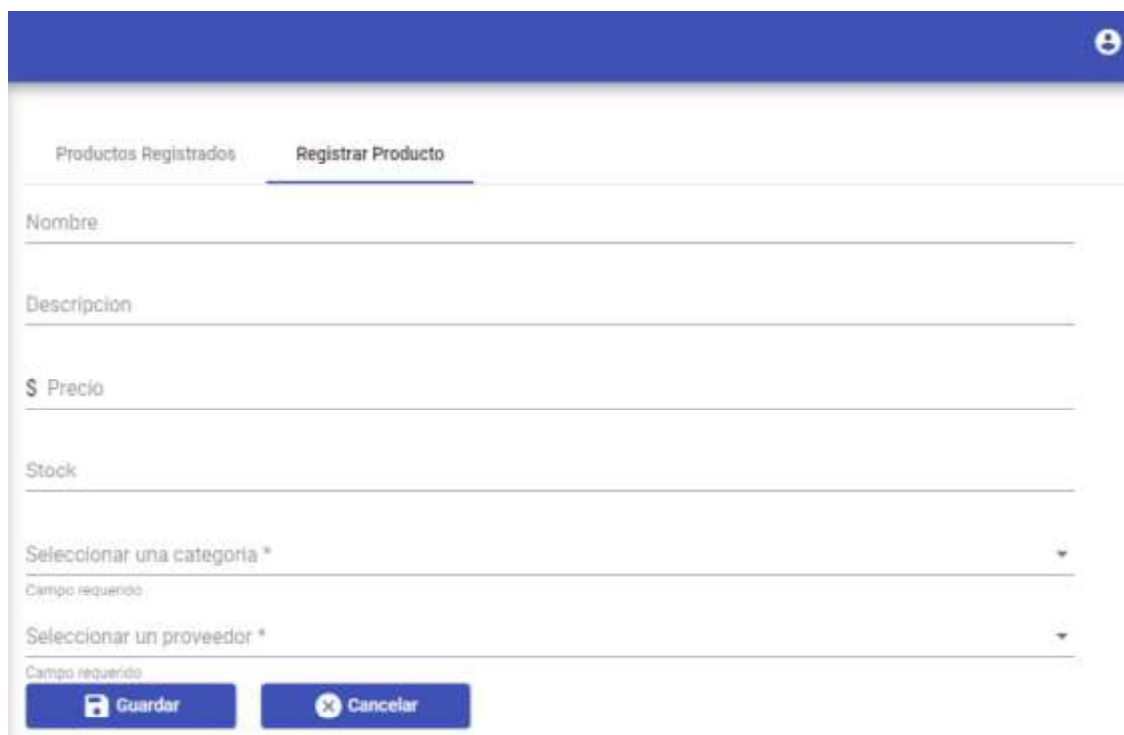


Figura 2.34 Interfaz gráfica del “*ProductComponent*”.

Por otra parte, el *componente de clase* de “*ProductComponent*” contiene el método *saveProduct()* mostrado en la línea 58 del Código 2.9. Este método recibe como argumento la información del formulario (*productForm*) definido en la *plantilla HTML* en la línea 58. El formulario es validado en la línea 59, asegurándose que todos los campos de tipo obligatorio han sido llenados. En la línea 61 se envía la información del nuevo producto hacia el servicio web, mediante en el método *subscribe()* se espera el resultado del servicio web ya que la petición es de tipo asincrónica. Una vez obtenida la respuesta de tipo satisfactorio del servidor se actualiza la lista de productos en la línea 62.

```
58 saveProduct(productForm: NgForm) {  
59   if (productForm.form.valid) {  
60     if (productForm.value._id == null) {  
61       this.productService.addProduct(productForm.value).subscribe(data => {  
62         this.getProducts();  
63         this.showMessage(data);  
64         this.resetForm(productForm);  
65         this.tabGroup.selectedIndex = this.PRODUCT_REGISTERED;  
66       });  
}
```

Código 2.9 Método *saveProduct()* en el componente de clase de “*ProductComponent*”.

2.3.5.2 Codificación de la Aplicación Móvil

Para la codificación de la aplicación móvil Android se utilizó el IDE oficial, Android Studio. La implementación se realizó siguiendo las guías de diseño del **Apartado 2.2.6**. Android Studio cuando crea un nuevo proyecto se encarga de generar archivos y carpetas que permitirán el desarrollo de la aplicación móvil. Es importante mencionar que la carpeta “*layout*” visualizada en la Figura 2.35, contiene los archivos que definirán las interfaces gráficas que componen el **Módulo Móvil** del prototipo de software distribuido.

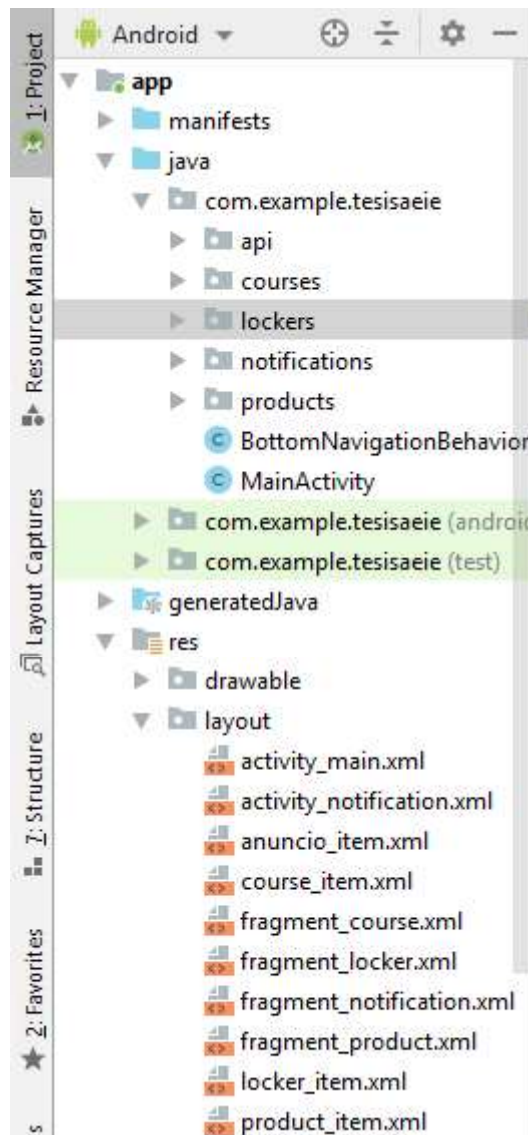


Figura 2.35 Interfaces de la aplicación móvil.

A continuación, se mostrará un ejemplo de código para la codificación de las interfaces gráficas, puesto que el archivo completo estará en el **ANEXO F**. Este anexo contendrá además todo el código de la aplicación móvil.

En el Código 2.10 se visualiza una parte del código para la implementación de las distintas interfaces gráficas. Android declara los distintos elementos que contendrá una interfaz de usuario a través de XML. Al usar vocabulario XML de Android, se puede crear rápidamente diseños de la interfaz de usuario y de los distintos elementos que la contienen

El Código 2.10 presenta en la línea 1, la declaración de un *CoordinatorLayout*, el cual permite organizar los elementos que conformarán la interfaz de usuario. En la línea 4 y 5, se observa las propiedades *layout_height* y *layout_width*. Estas propiedades indican que la interfaz tomará el largo y ancho completo de la pantalla del celular donde la aplicación sea instalada. En la línea 10 se observa un *FrameLayout* que funciona como un contenedor para las distintas interfaces que componen el **Módulo Móvil**. Por otra parte, en la línea 17 se define un menú mediante un *BottomNavigationView* que permitirá la navegación. La ubicación del menú se define en la línea 21, indicando que se encontrará en la parte inferior de la pantalla.

```
activity_main.xml x
1  <?xml version="1.0" encoding="utf-8" ?>
2  <android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res-auto"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      tools:context=".MainActivity"
8      android:background="@color/background">
9
10     <FrameLayout
11         android:id="@+id/container"
12         android:layout_width="match_parent"
13         android:layout_height="match_parent"
14         app:layout_behavior="android.support.design.widget.AppBarLayout$ScrollingViewBehavior"
15     ></FrameLayout>
16
17     <android.support.design.widget.BottomNavigationView
18         android:id="@+id/nav_view"
19         android:layout_width="match_parent"
20         android:layout_height="wrap_content"
21         android:layout_gravity="bottom"
22         android:layout_marginStart="0dp"
23         android:layout_marginEnd="0dp"
24         android:background="?android:attr/windowBackground"
25         app:layout_constraintBottom_toBottomOf="parent"
26         app:layout_constraintLeft_toLeftOf="parent"
27         app:layout_constraintRight_toRightOf="parent"
28         app:menu="@menu/bottom_nav_menu" />
29
30 </android.support.design.widget.CoordinatorLayout>
```

Código 2.10 Layout para la interfaz gráfica.

En la Figura 2.36 se presenta las interfaces implementadas del módulo Móvil, las cuales corresponden a las opciones “*Productos*” y “*Casilleros*” dentro del menú ubicado en la parte inferior de la pantalla. Dentro de “*Productos*” el estudiante podrá visualizar por mediante una lista los productos ofertados por la AEIE, cada producto muestra las siguientes propiedades: nombre del producto, valor unitario del producto y *stock*.

Por otra parte, en “*Casilleros*” el Estudiante estará en capacidad de visualizar si existen casilleros disponibles para alquilarlos, cada casillero tiene su identificador representado a través de una letra y un número por ejemplo: A1. Con el fin de presentar una interfaz amigable, se ha utilizado gráficos de candados con color rojo y verde, que indican si el estado del casillero es “OCUPADO” o “LIBRE” respectivamente.



Figura 2.36 Interfaz módulo Móvil correspondiente a los menú Cursos y Anuncios.

En la Figura 2.37 se presenta las interfaces implementadas del módulo Móvil, las cuales corresponden a las opciones “Cursos” y “Anuncios” dentro del menú ubicado en la parte inferior de la pantalla. Dentro de “Cursos” el Estudiante podrá visualizar por medio de una lista los cursos ofertados por la AEIE. Cada curso muestra las siguientes propiedades: nombre, costo, cupos disponibles, fecha de inicio y semestre en el que se el curso es dictado.

En cambio, en “Anuncios” el Estudiante estará en capacidad de visualizar los anuncios de tipo informativo publicados por la AEIE. La información que se publique será a discreción de la AEIE y administrada mediante el módulo “Informativo” correspondiente a la aplicación web. Los anuncios son organizados a través de una lista, cada anuncio constan de un título, la descripción con la información que se presentará al estudiante, la fecha y hora de la publicación del anuncio.

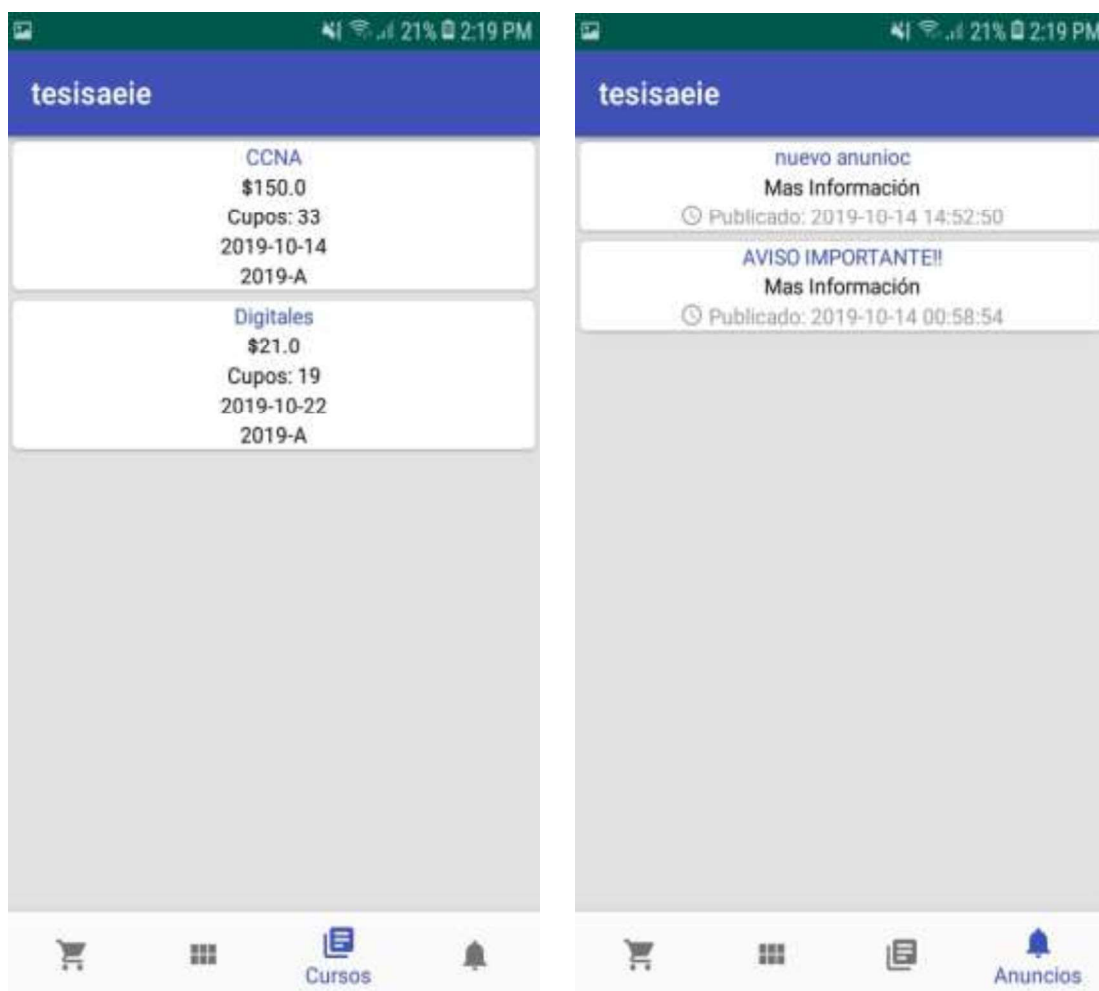


Figura 2.37 Interfaz módulo Móvil correspondiente a los menú Cursos y Anuncios.

3 RESULTADOS Y DISCUSIÓN

En este capítulo se realizarán las pruebas sobre el prototipo implementado. Se verificará el cumplimiento de Requerimientos Funcionales y Requerimientos No Funcionales definidos en el **Apartado 2.2.2**. Además, se analizarán los resultados obtenidos.

3.1 ACTUALIZACIÓN TABLERO KANBAN

En la Figura 3.1 se visualiza el tablero actualizado con las tareas correspondientes para la etapa de pruebas del prototipo de software.



Figura 3.1 Tablero Kanban para la fase de pruebas.

3.2 PRUEBAS DE VALIDACIÓN DE REQUERIMIENTOS FUNCIONALES

En este apartado se valida el adecuado funcionamiento de los Requerimientos Funcionales de cada uno de los módulos que componen el prototipo de software distribuido descritos en el **Apartado 2.2.5**. Para comenzar se detallan los RF de cada módulo y posteriormente se procede con la validación de los mismos.

3.2.1 PRUEBA DE VALIDACIÓN DEL MÓDULO CATÁLOGO

Este módulo permite la gestión de los productos ofertados por la AEIE. La Tabla 3.1 muestra los RF por cumplir en este módulo.

Tabla 3.1 Requerimientos Funcionales del Módulo Catálogo.

CASE DE USO	RF	DESCRIPCIÓN
CU01	RF01	Añadir categoría de productos.
CU01	RF02	Modificar categoría de productos.
CU01	RF03	Eliminar categoría de productos.
CU01	RF04	Añadir productos.
CU01	RF05	Modificar productos.
CU01	RF06	Eliminar productos.
CU01	RF07	Añadir proveedor de productos.
CU01	RF08	Modificar proveedor productos.
CU01	RF09	Eliminar proveedor de productos.

Como se observa en la Figura 3.2, una vez registrado el nombre de la categoría se visualiza un mensaje de confirmación o un mensaje de error si se produce un error. De esta manera se demuestra el cumplimiento del RF01.

The screenshot shows a web interface for 'Registrar Categoría'. It has two tabs: 'Categorías Registradas' and 'Registrar Categoría'. The 'Registrar Categoría' tab is active. There are two input fields: 'Nombre' with the text 'bebidas' and 'Descripcion'. At the bottom, there are two buttons: 'Guardar' and 'Cancelar'. On the right side, there are two dark blue notification boxes. The top one says 'Categoría creada exitosamente' with an 'OK' button. The bottom one says 'Se produjo un error. Intentelo nuevamente' with an 'OK' button.

Figura 3.2 Creación de una categoría.

Si se desea actualizar la información de la categoría, hay que dar *click* en el icono “Editar” con la figura de un lápiz mostrado en la Figura 3.3.



Figura 3.3 Acciones Editar y Eliminar.

Después de presionar el icono “Editar”, se ingresa el nuevo valor y se pulsa el botón “Guardar”. Finalmente, en la Figura 3.4 se comprueba el cumplimiento del RF02, puesto que se observa un mensaje de información indicando que la actualización ha sido exitosa.



Figura 3.4 Mensaje de actualización exitosa para una categoría.

Por otra parte si se desea eliminar una categoría, se selecciona el icono “Eliminar” con la figura de un basurero mostrado previamente en la Figura 3.3. Después de pulsar el icono, aparece un cuadro de diálogo para confirmar la eliminación de la categoría. El cumplimiento del RF03 se presenta en la Figura 3.5.

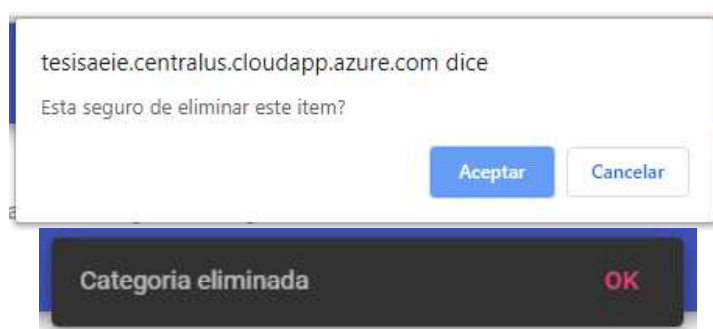
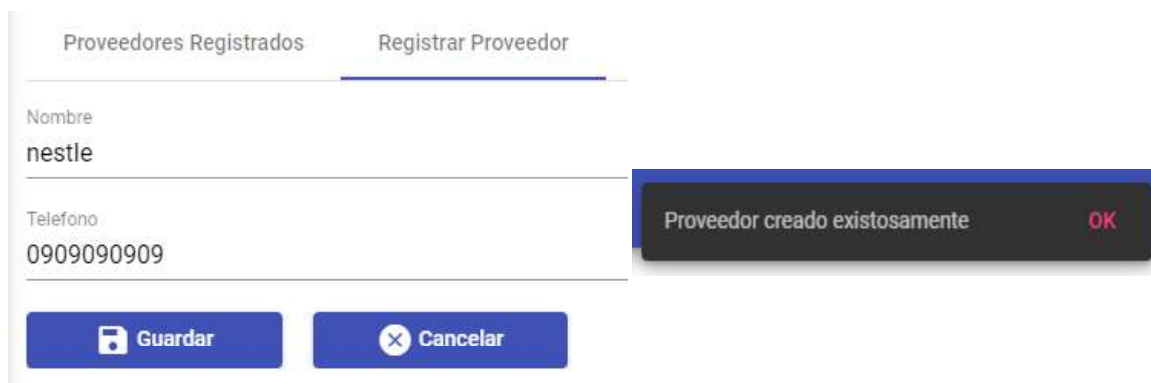


Figura 3.5 Mensaje de eliminación exitosa para una categoría.

Como se observa en la Figura 3.6, una vez registrado el proveedor se visualiza un mensaje de confirmación, este mensaje indica que el proveedor se registró exitosamente. De esta manera se demuestra el cumplimiento del RF07.



The screenshot shows a web interface with two tabs: 'Proveedores Registrados' and 'Registrar Proveedor'. The 'Registrar Proveedor' tab is active. Below the tabs, there are two input fields: 'Nombre' with the value 'nestle' and 'Telefono' with the value '0909090909'. To the right of these fields, a dark blue notification box displays the text 'Proveedor creado existosamente' in white, with a red 'OK' button. At the bottom of the form, there are two blue buttons: 'Guardar' (with a save icon) and 'Cancelar' (with a close icon).

Figura 3.6 Creación de un proveedor.

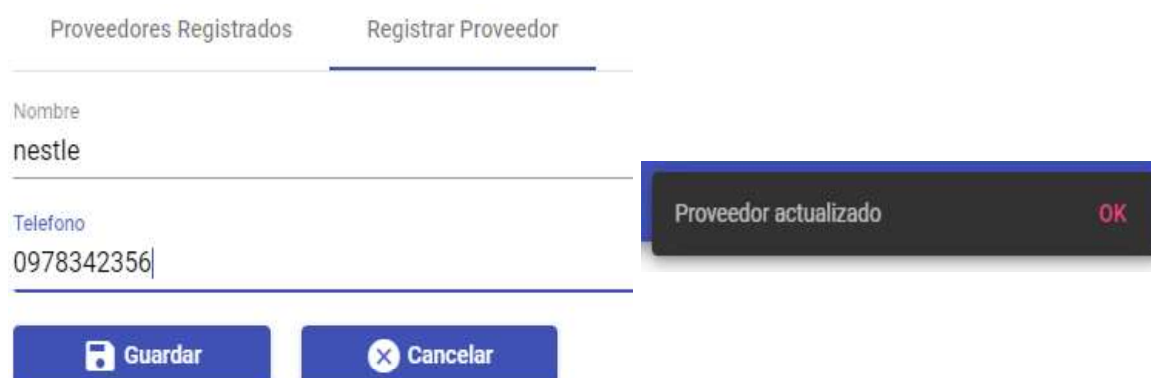
Si se desea actualizar la información de un proveedor, hay que dar *click* en el icono “Editar” con la figura de un lápiz mostrado en la Figura 3.7.



Nombre	Telefono	Actualizar	Eliminar
nestle	0909090909		

Figura 3.7 Acciones de eliminar y actualizar.

Después de presionar el icono “Editar”, se ingresa el nuevo valor y se pulsa el botón “Guardar”. Finalmente, en la Figura 3.8 se comprueba el cumplimiento del RF08, puesto que se observa un mensaje de información indicando que la actualización ha sido exitosa.



The screenshot shows the same web interface as Figure 3.6, but with the 'Actualizar' tab active. The 'Nombre' field still contains 'nestle', but the 'Telefono' field now contains '0978342356'. The notification box on the right now displays 'Proveedor actualizado' in white, with a red 'OK' button. The 'Guardar' and 'Cancelar' buttons remain at the bottom.

Figura 3.8 Mensaje de actualización exitosa para un proveedor.

Por otra parte, si se desea eliminar un proveedor, se selecciona el icono “Eliminar” con la figura de un basurero mostrado previamente en la Figura 3.7. Después de pulsar el icono, aparece un cuadro de diálogo para confirmar la eliminación del proveedor. El cumplimiento del RF09 se presenta en la Figura 3.9.

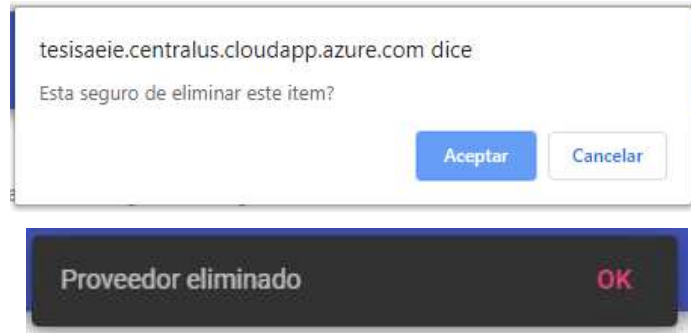


Figura 3.9 Mensaje de eliminación exitosa para un proveedor.

Como se observa en la Figura 3.10, una vez registrado el producto se visualiza un mensaje de confirmación, este mensaje indica que el producto se registró exitosamente. De esta manera se demuestra el cumplimiento del RF04. Si existe un problema se observa un mensaje de error.

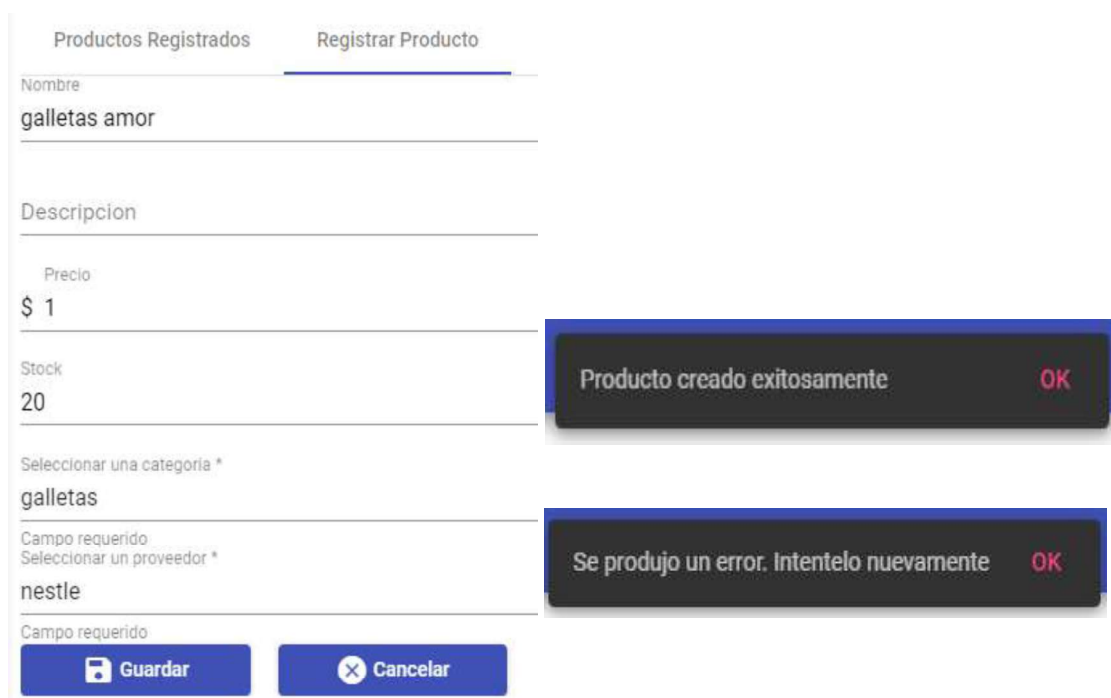


Figura 3.10 Creación de un producto.

Si se desea actualizar la información de un producto, hay que dar *click* en el icono “Editar” con la figura de un lápiz mostrado en la Figura 3.11.

Nombre	Descripción	Precio	Stock	Categoría	Proveedor	Actualizar	Eliminar
galletas amor		\$1.00	20	galletas	nestle		

Figura 3.11 Acciones de eliminar y actualizar.

Después de presionar el icono “Editar”, se ingresa el nuevo valor y se pulsa el botón “Guardar”. Finalmente, en la Figura 3.12 se comprueba el cumplimiento del RF05, puesto que se observa un mensaje de información indicando que la actualización ha sido exitosa.



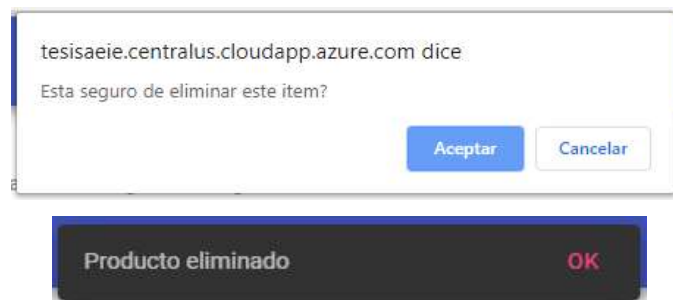
The screenshot shows a web form titled "Registrar Producto" with the following fields:

- Nombre: galletas amor
- Descripción: (empty)
- Precio: \$ 1
- Stock: 10
- Seleccionar una categoría *: galletas
- Campo requerido Seleccionar un proveedor *: nestle

 At the bottom, there are "Guardar" and "Cancelar" buttons. A dark blue notification box on the right displays the message "Producto actualizado" with an "OK" button.

Figura 3.12 Mensaje de actualización exitosa para un producto.

Por otra parte si se desea eliminar un producto, se selecciona el icono “Editar” con la figura de un basurero mostrado previamente en la Figura 3.11. Después de pulsar el icono, aparece un cuadro de diálogo para confirmar la eliminación del producto. El cumplimiento del RF06 se presenta en la Figura 3.13.



The screenshot shows a confirmation dialog box with the text:

tesisaeie.centralus.cloudapp.azure.com dice
Esta seguro de eliminar este item?

 Below the text are "Aceptar" and "Cancelar" buttons. Below the dialog box, a dark blue notification box displays the message "Producto eliminado" with an "OK" button.

Figura 3.13 Mensaje de eliminación exitosa de un producto.

3.2.2 PRUEBA DE VALIDACIÓN DEL MÓDULO VENTAS

En este módulo se verificará la gestión de las ventas de la AEIE. La Tabla 3.2 muestra los RF que se deben cumplir.

Tabla 3.2 Requerimientos Funcionales del Módulo Ventas.

CASO DE USO	RF	DESCRIPCIÓN
CU02	RF10	Registrar ventas del día.
CU02	RF11	Listar todas las ventas.

Cuando un usuario desee realizar una venta podrá seleccionar un producto de la lista de productos registrados en el sistema, como se observa en la Figura 3.14.

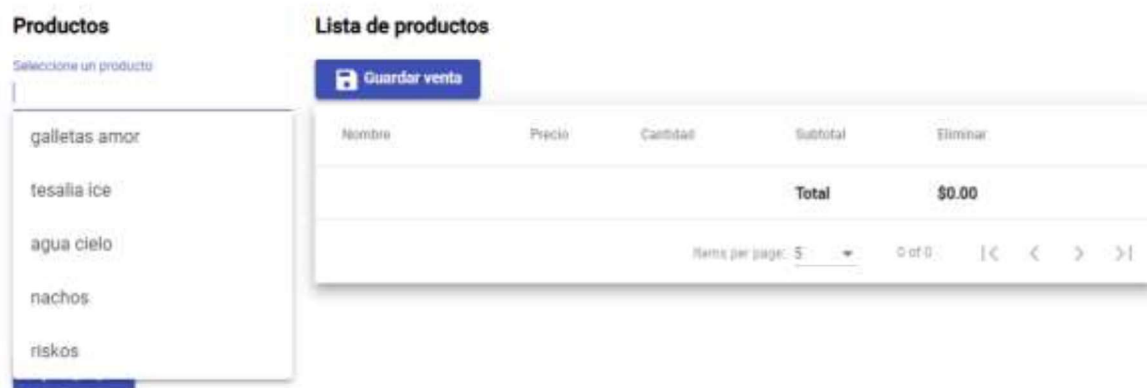


Figura 3.14 Selección de un producto para la venta.

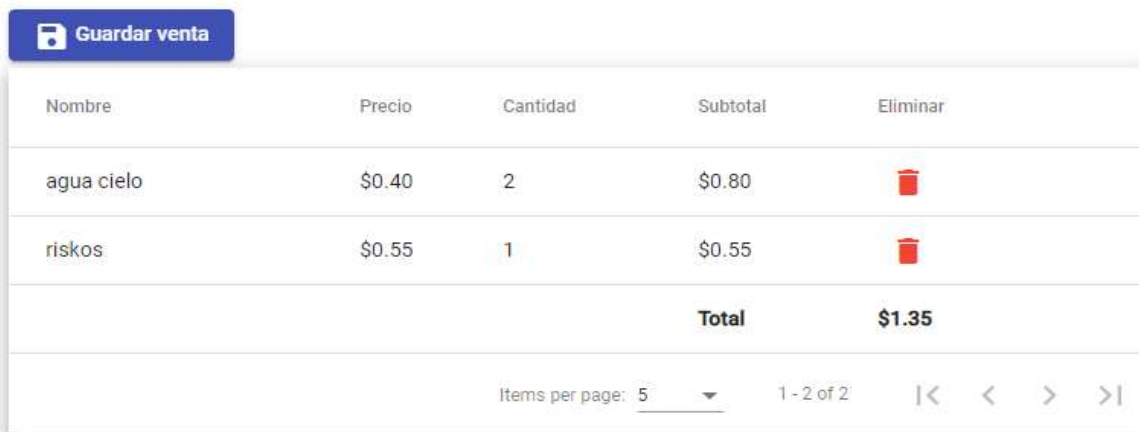
Si el usuario no ha seleccionado un producto y presiona el botón “Agregar”, se presentará el mensaje “Ingrese algún producto” indicando que es necesario añadir un producto para que pueda ser agregado a la lista de productos por venderse. Además, si un producto fue seleccionado pero no se ha indicado la cantidad a venderse, se mostrará un mensaje al usuario aclarando que debe ingresar un valor diferente de cero. En la Figura 3.15 se observa los mensajes de información mencionados anteriormente que se exhibirán al usuario.



Figura 3.15 Mensaje de información al usuario.

La Figura 3.16 muestra una lista con los productos que se registraran cuando se pulse el boton "Guardar venta". Además, si se desea eliminar un producto de la lista de productos basta con dar click en el icono de color rojo en la columna "Eliminar".

Lista de productos



The screenshot shows a blue button labeled "Guardar venta" with a save icon. Below it is a table with the following data:

Nombre	Precio	Cantidad	Subtotal	Eliminar
agua cielo	\$0.40	2	\$0.80	
riskos	\$0.55	1	\$0.55	
			Total	\$1.35

At the bottom of the table, there is a pagination control showing "Items per page: 5" and "1 - 2 of 2" with navigation arrows.

Figura 3.16 Venta de productos.

Finalmente cuando se presiona el botón "Guardar venta" mostrado en la Figura 3.16, se presentará un mensaje al usuario indicando que la venta se registró con éxito; probando así el cumplimiento del RF10. El mensaje mencionado anteriormente se visualiza en la Figura 3.17.

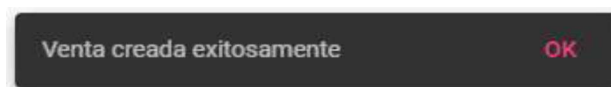


Figura 3.17 Mensaje de registro de venta exitoso.

La Figura 3.18 valida el cumplimiento del RF11, puesto que se observa la lista de ventas registradas. Cada venta contiene información acerca de qué usuario realizó la venta, la hora y fecha de la venta, los productos y la cantidad de los mismos que fueron vendidos, y el valor total de la venta.



The screenshot shows a table titled "Ventas registradas" with the following data:

Usuario	Fecha	Productos Vendidos	Total
verito	10/30/19, 1:27 AM	2 - Ver detalle 2 - agua cielo 1 - riscos	\$1.35

At the bottom, there is a pagination control showing "Items per page: 5" and "1 - 1 of 1" with navigation arrows.

Figura 3.18 Lista de ventas registradas.

3.2.3 PRUEBA DE VALIDACIÓN DEL MÓDULO SERVICIOS

Este módulo permite la gestión de los servicios ofertados por la AEIE. La Tabla 3.3 muestra los RF por cumplir en este módulo.

Tabla 3.3 Requerimientos Funcionales del Módulo Servicios.

CASO DE USO	RF	DESCRIPCIÓN
CU03	RF12	Registrar semestre.
CU03	RF13	Añadir estudiantes.
CU03	RF14	Modificar estudiantes.
CU03	RF15	Eliminar estudiantes.
CU03	RF16	Registrar el aporte de un estudiante.
CU03	RF17	Eliminar el aporte de un estudiante.
CU03	RF18	Añadir casilleros.
CU03	RF19	Modificar casilleros.
CU03	RF20	Eliminar casilleros.
CU03	RF21	Añadir el alquiler de casillero por un estudiante.
CU03	RF22	Eliminar el registro de un estudiante que alquila un casillero.
CU03	RF23	Notificar mediante correo electrónico el estado de alquiler del casillero al estudiante.

Como se observa en la Figura 3.19, el formulario mostrado permite el registro de un semestre referencial para las actividades llevadas a cabo por la AEIE. Una vez registrado el semestre al presionar el botón “Guardar”, se visualiza un mensaje de confirmación. De esta manera se demuestra el cumplimiento del RF12. Si existo un error muestra un mensaje de error al usuario.



Figura 3.19 Registro de un semestre referencial.

Como se observa en la Figura 3.20, una vez que se ha llenado todos los campos se procede a guardar el registro del estudiante pulsando el botón “Guardar”. Seguidamente se visualiza un mensaje de confirmación, de esta manera se demuestra el cumplimiento del RF13.

The screenshot shows a web interface for registering a student. It has two tabs: "Estudiantes Registrados" and "Registrar Estudiante", with the second tab selected. The form contains the following fields:

- Nombres: Ricardo David
- Apellidos: Taipe Hidalgo
- Selección de carrera: Redes
- Cedula: 1722241666
- Numero unico: 201210674
- Email: rdavid28@hotmail.es
- Telefono: 0987762011

At the bottom of the form are two buttons: "Guardar" (Save) and "Cancelar" (Cancel). A dark grey notification box with a white border is overlaid on the right side of the form, containing the text "Estudiante creado exitosamente" and an "OK" button.

Figura 3.20 Creación de una categoría.

Si se desea actualizar la información de un estudiante, hay que dar *click* en el icono “Editar” con la figura de un lápiz mostrado en la Figura 3.21.

Nombres	Apellidos	Carrera	Cedula	Numero Unico	Correo	Telefono	Actualizar	Eliminar
Ricardo David	Taipe Hidalgo	Redes	1722241666	201210674	rdavid28@hotmail.es	0987762011		

Figura 3.21 Acciones Editar y Eliminar.

Después de presionar el icono “Editar”, se ingresa el nuevo valor y se pulsa el botón “Guardar”. Finalmente, en la Figura 3.22 se comprueba el cumplimiento del RF14, puesto que se observa un mensaje de información indicando que la actualización fue exitosa.

The image shows a web form for registering a student. The form has two tabs: 'Estudiantes Registrados' and 'Registrar Estudiante'. The 'Registrar Estudiante' tab is active. The form fields are as follows:

- Nombres:** Ricardo David
- Apellidos:** Taípe Hidalgo
- Seleccionar una carrera *:** Redes
- Cedula:** 1722241666
- Numero unico:** 201210674
- Email:** ricardo.taípe@epn.edu.ec
- Telefono:** 0987762011

At the bottom of the form are two buttons: 'Guardar' (with a save icon) and 'Cancelar' (with a close icon). A dark blue success message box is overlaid on the right side of the form, containing the text 'Estudiante actualizado' and a red 'OK' button.

Figura 3.22 Mensaje de actualización exitosa.

Por otra parte si se desea eliminar un estudiante, se selecciona el icono “Eliminar” con la figura de un basurero mostrado previamente en la Figura 3.21. Después de pulsar el icono, aparece un cuadro de diálogo para confirmar la eliminación del estudiante. El cumplimiento del RF15 se presenta en la Figura 3.23.

The image shows a confirmation dialog box. The text inside the dialog box reads: 'tesisaeie.centralus.cloudapp.azure.com dice Esta seguro de eliminar este item?'. Below the text are two buttons: 'Aceptar' (Accept) and 'Cancelar' (Cancel). Below the dialog box is a dark blue success message box containing the text 'Estudiante eliminado' and a red 'OK' button.

Figura 3.23 Mensaje de eliminación exitosa para una categoría.

En la Figura 3.24 se observa el formulario que permite registrar del aporte realizado por un estudiante. Para guardar el registro se selecciona el semestre en que se realiza el aporte y se busca el estudiante mediante una lista que se presenta al usuario.

Aportes Registrados Registrar Aporte

Seleccionar un semestre *
2019-A

Buscar Estudiante *
5db9385c5df6f758ce28b1e4

RICARDO DAVID TAIPE HIDALGO
Ingrese el valor del aporte
\$ 20

Descripcion

Fecha
Fri Nov 01 2019 19:00:08 GMT-0500 (hora de Ecuador)

 Guardar  Cancelar

Figura 3.24 Registro de aporte.

Cuando se han llenado todos los campos del formulario en la Figura 3.24, se procede a guardar la información. La Figura 3.25 muestra el cumplimiento del RF16.

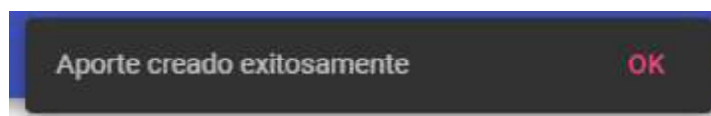




Figura 3.25 Mensaje de registro exitoso del aporte.

Para eliminar un aporte se debe hacer *click* en el icono rojo en la columna “Eliminar”. Una vez completada la acción el usuario visualizará el mensaje de la Figura 3.26, dando cumplimiento al RF17.

Nombres	Apellidos	Semestre	Fecha	Valor	Descripcion	Actualizar	Eliminar
Ricardo David	Taipe Hidalgo	2019-A	11/1/19, 7:00 PM	\$20.00			

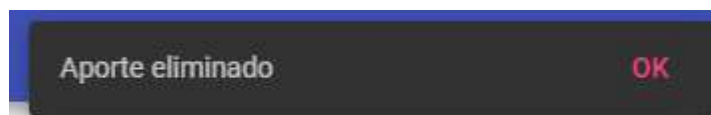


Figura 3.26 Eliminación de un aporte.

Como se observa en la Figura 3.2, una vez registrado el identificador del casillero, letra y número, se visualiza un mensaje de confirmación. De esta manera se demuestra el cumplimiento del RF18.



Figura 3.27 Creación de un casillero.

Si se desea actualizar la información de un casillero, hay que dar *click* en el icono “Editar” con la figura de un lápiz mostrado en la Figura 3.28.



Descripcion	Estado Casillero	Actualizar	Eliminar
A1	LIBRE		

Figura 3.28 Acciones Editar y Eliminar.

Después de presionar el icono de Editar, se ingresa el valor a modificar y se pulsa el botón “Guardar”. Finalmente, en la Figura 3.29 se comprueba el cumplimiento del RF19, puesto que se observa un mensaje de información indicando que la actualización fue exitosa.



Figura 3.29 Mensaje de actualización exitosa para una categoría.

Por otra parte si se desea eliminar un casillero, se selecciona el icono “Eliminar” con la figura de un basurero mostrado previamente en la Figura 3.28. Después de pulsar el icono, aparece un cuadro de diálogo para confirmar la eliminación del casillero. El cumplimiento del RF20 se presenta en la Figura 3.30.

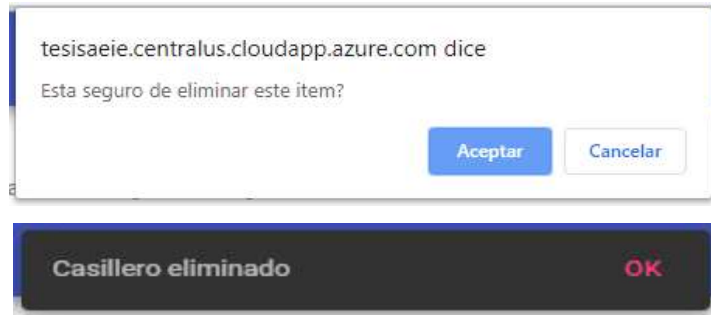


Figura 3.30 Mensaje de eliminación exitosa de un casillero.

La Figura 3.31 evidencia el cumplimiento del RF21. Primero se selecciona el casillero, después se selecciona el estudiante que alquilará el casillero. Finalmente, se registra el precio del alquiler y se guarda el registro.

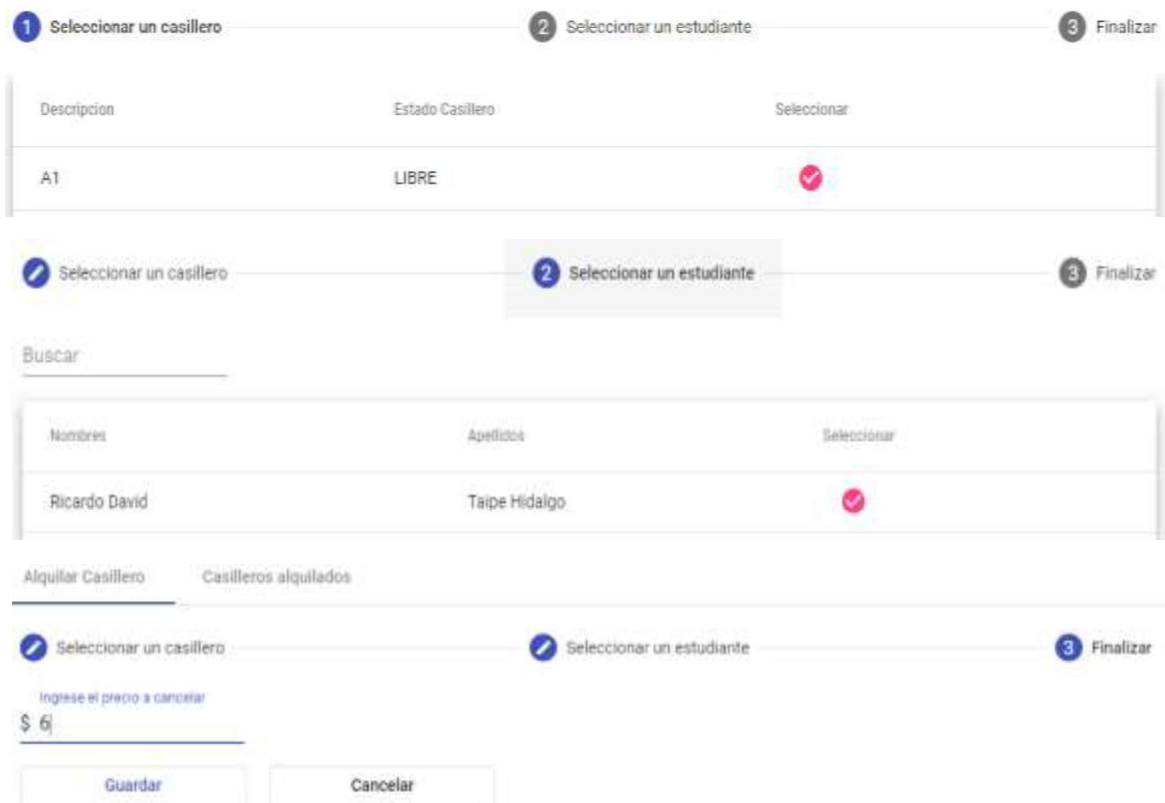


Figura 3.31 Alquiler de un casillero.

Terminado el proceso, el usuario visualiza un mensaje de confirmación mostrado en la Figura 3.32

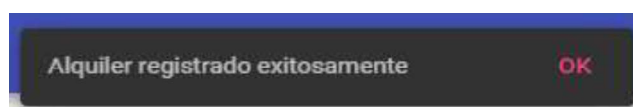


Figura 3.32 Mensaje de alquiler exitoso.

La Figura 3.33 muestra las acciones que se puedan ejecutar sobre un casillero que ha sido alquilado por un estudiante, estas acciones son: “Notificar retiro de pertenencias personales” y “Terminar el alquiler”.

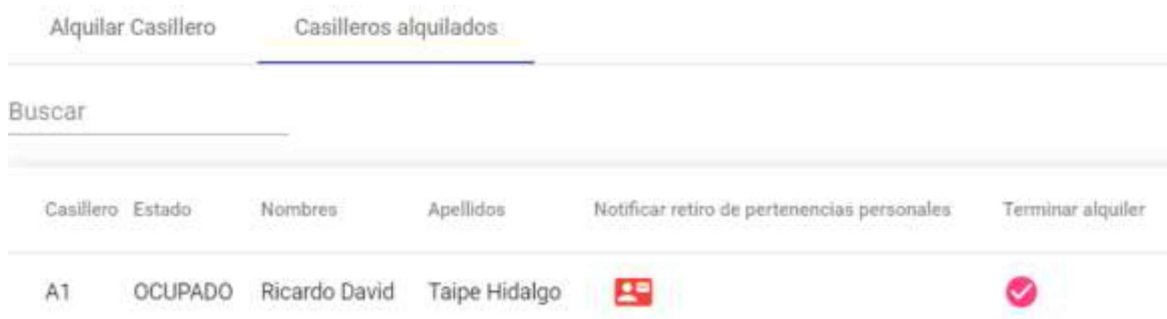


Figura 3.33 Acciones disponibles sobre un casillero alquilado.

Después de presionar el botón de color rojo encargado de enviar la notificación de retiro de pertenencias personales al estudiante, se presenta un mensaje de confirmación al usuario indicado en la Figura 3.34

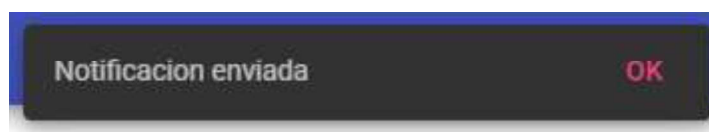


Figura 3.34 Mensaje de notificación enviada con éxito.

En la Figura 3.35 se puede observar el correo que recibe el estudiante, indicando que debe acercarse a retirar sus pertenencias personales en caso de no renovar el alquiler. De esta manera se verifica el cumplimiento del RF23.

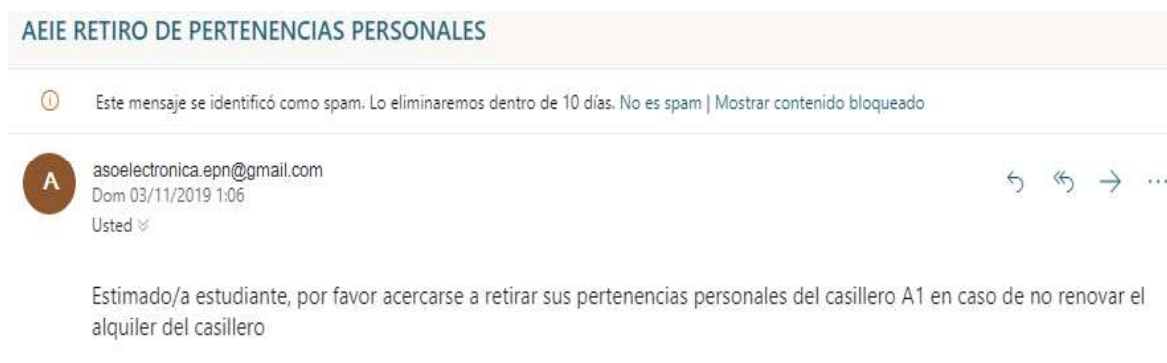


Figura 3.35 Correo electrónico que recibe el estudiante.

La Figura 3.36 da cumplimiento al RF22. El mensaje que se visualiza es presentado al usuario cuando se pulsa el botón que da por terminado el alquiler de casillero.



Figura 3.36 Eliminación de alquiler.

3.2.4 PRUEBA DE VALIDACIÓN DEL MÓDULO CURSOS

En este módulo se verificará el funcionamiento de la gestión de los cursos ofertados por la AEIE. La Tabla 3.4 muestra los RF que se deben cumplir para este módulo.

Tabla 3.4 Requerimientos Funcionales del Módulo Cursos.

CASO DE USO	RF	DESCRIPCIÓN
CU04	RF24	Añadir cursos ofertados por la AEIE.
CU04	RF25	Modificar cursos ofertados por la AEIE.
CU04	RF26	Eliminar cursos ofertados por la AEIE.
CU04	RF27	Registrar estudiantes en los cursos.

Como se observa en la Figura 3.37, una vez registrado los datos del nuevo curso se visualiza un mensaje de confirmación. De esta manera se demuestra el cumplimiento del RF24.



Figura 3.37 Creación de un curso.

Si se desea actualizar la información de la categoría, hay que dar *click* en el icono “Editar” con la figura de un lápiz mostrado en la Figura 3.38.

Nombre	Descripción	Precio	Cupos	Fecha	Semestre	Actualizar	Eliminar
Curso CCNA 1 CGO OFICIAL		\$168.00	20	11/21/19	2019-B		

Figura 3.38 Acciones Editar y Eliminar.

Después de presionar el icono “Editar”, se ingresa el valor a modificarse y se pulsa el botón “Guardar”. Finalmente, en la Figura 3.39 se comprueba el cumplimiento del RF25, puesto que se observa un mensaje de información indicando que la actualización ha sido exitosa.

The image shows a form titled "Registrar Curso" with the following fields:

- Nombre: Curso CCNA 1 CGO OFICIAL
- Descripción: (empty)
- Precio: \$ 168
- Fecha: 11/21/2019 (with a calendar icon and a note "Seleccione una fecha en el icono de la izquierda *")
- Cupos: 30
- Semestre: 2019-B (with a note "Seleccione un semestre *")

 At the bottom are "Guardar" and "Cancelar" buttons. A dark blue overlay with white text says "Curso actualizado" and a red "OK" button is visible on the right side.

Figura 3.39 Mensaje de actualización exitosa para un curso.

Por otra parte si se desea eliminar un curso, se selecciona el icono “Eliminar” con la figura de un basurero mostrado previamente en la Figura 3.38. Después de pulsar el icono, aparece un cuadro de diálogo para confirmar la eliminación del curso. El cumplimiento del RF26 se presenta en la Figura 3.40.

The image shows a confirmation dialog box with the text:

- tesisaie.centralus.cloudapp.azure.com dice
- Esta seguro de eliminar este item?

 There are "Aceptar" and "Cancelar" buttons. Below the dialog is a dark blue overlay with white text saying "Categoría eliminada" and a red "OK" button.

Figura 3.40 Mensaje de eliminación exitosa para una categoría.

La Figura 3.41 muestra el cumplimiento del RF27. Primero se selecciona el curso, luego el estudiante y finalmente se guarda el registro del estudiante en el curso.

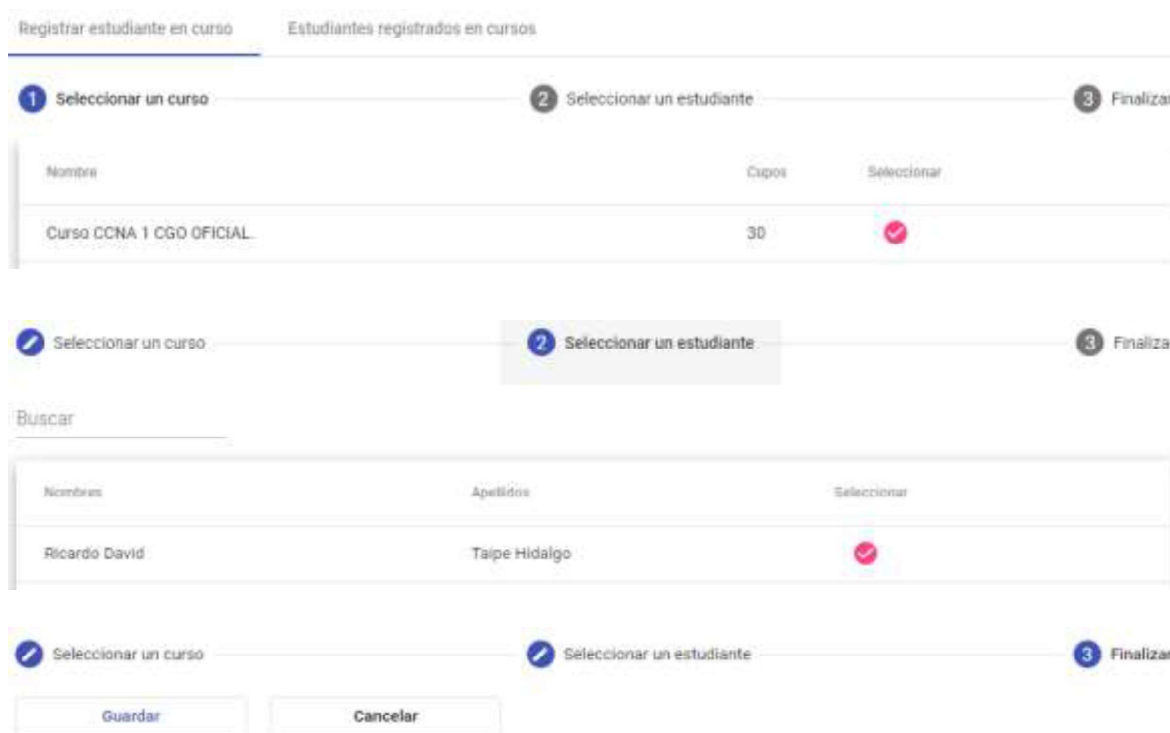


Figura 3.41 Registro de un estudiante en un curso.

3.2.5 PRUEBA DE VALIDACIÓN DEL MÓDULO INFORMATIVO

En este módulo se verificará la gestión de los avisos informativos publicados por la AEIE. La Tabla 3.5 muestra los RF que se deben cumplir para este módulo.

Tabla 3.5 Requerimientos Funcionales del Módulo Informativo.

CASO DE USO	RF	DESCRIPCIÓN
CU05	RF28	Publicar eventos, avisos de tipo informativo que serán visualizados en la aplicación móvil destinada a los estudiantes.
CU05	RF29	Modificar avisos publicados.
CU05	RF30	Eliminar avisos publicados.

Como se observa en la Figura 3.42, una vez registrado el título y la descripción del anuncio se procede a guardar; visualizando un mensaje de confirmación. De esta manera se demuestra el cumplimiento del RF28.

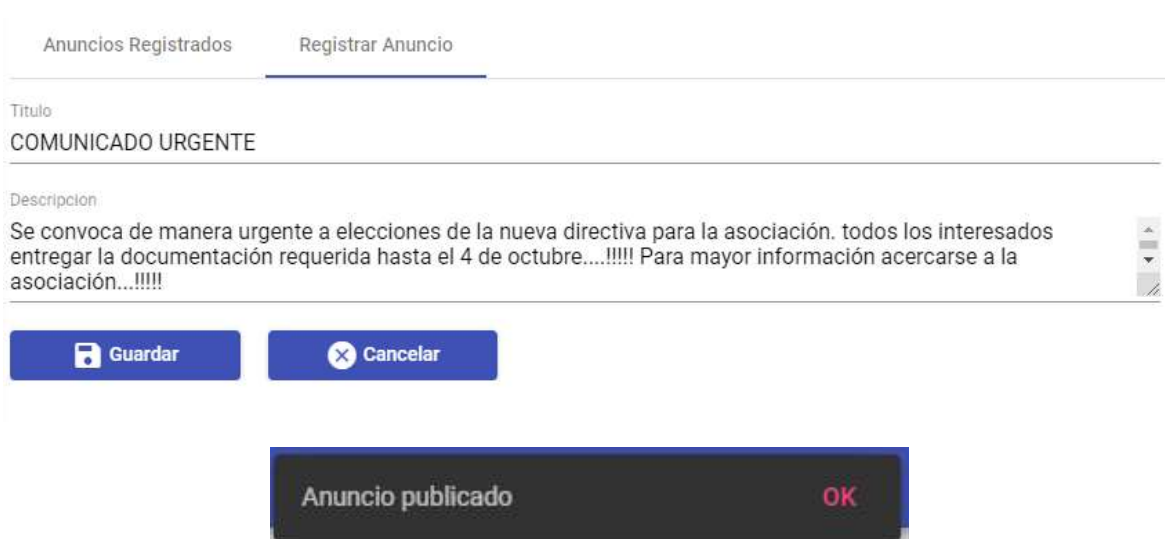


Figura 3.42 Publicación de un anuncio de tipo informativo.

Si se desea actualizar la información de un anuncio, hay que dar *click* en el icono “Editar” con la figura de un lápiz mostrado en la Figura 3.43.

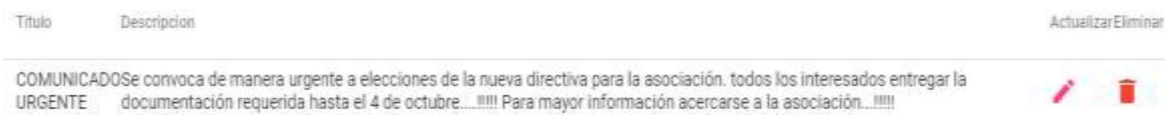


Figura 3.43 Acciones Editar y Eliminar.

Después de presionar el icono “Editar”, se ingresa el valor a modificarse y se pulsa el botón “Guardar”. Finalmente, en la Figura 3.44 se comprueba el cumplimiento del RF29, puesto que se observa un mensaje de información indicando que la actualización ha sido exitosa.

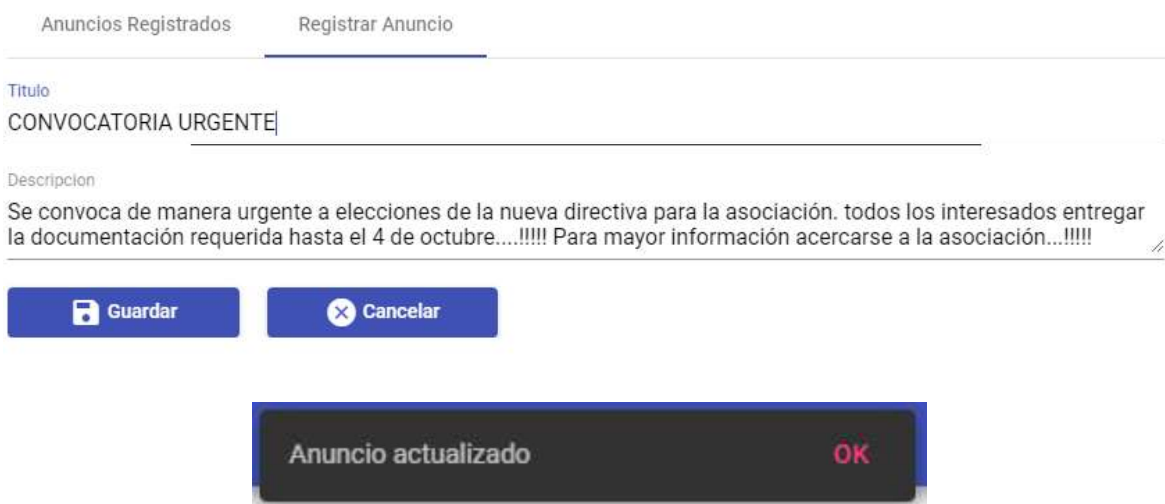


Figura 3.44 Mensaje de actualización exitosa para un anuncio.

Por otra parte si se desea eliminar un anuncio, se selecciona el icono “Eliminar” con la figura de un basurero mostrado previamente en la Figura 3.43. Después de pulsar el icono, aparece un cuadro de diálogo para confirmar la eliminación del anuncio. El cumplimiento del RF30 se presenta en la Figura 3.45.

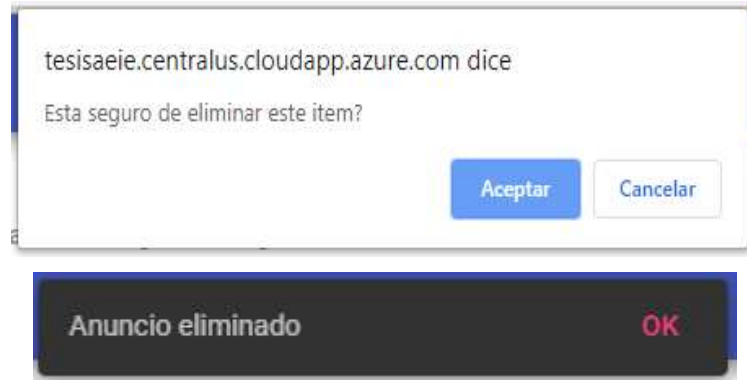


Figura 3.45 Mensaje de eliminación exitosa para un anuncio.

3.2.6 PRUEBA DE VALIDACIÓN DEL MÓDULO USUARIOS

Este módulo permite la gestión de los usuarios pertenecientes a la AEIE encargados de la administración de la aplicación web. La Tabla 3.6 muestra los RF por cumplir en este módulo.

Tabla 3.6 Requerimientos Funcionales del Módulo Usuarios.

CASO DE USO	RF	DESCRIPCIÓN
CU06	RF31	Registrar usuario.
CU06	RF32	Modificar usuario.
CU06	RF33	Eliminar usuario.
CU06	RF34	Listar usuarios.

Como se observa en la Figura 3.46, una vez registrado todos los datos del formulario se procede a guardar; visualizando un mensaje de confirmación. De esta manera se demuestra el cumplimiento del RF31. Cabe mencionar que la propiedad “Admin” presente en la Figura 3.46, permitirá indicar si un usuario que se está creando será de tipo administrador. Un usuario de tipo “Administrador” será el único que tendrá acceso al módulo usuarios, caso contrario si indicará al usuario a través de un mensaje, como indica la Figura 3.47 , que no tiene permiso para utilizar dicho módulo.

Figura 3.46 Creación de un nuevo usuario.

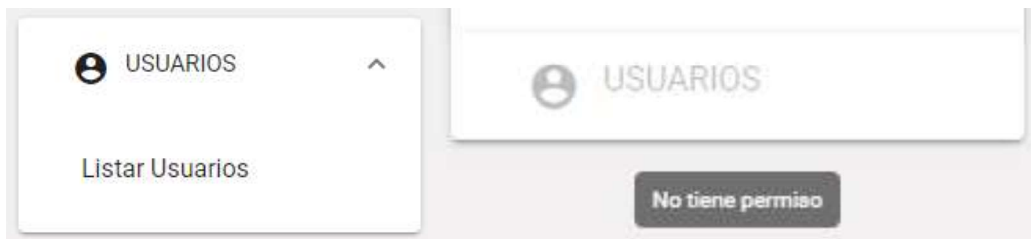


Figura 3.47 Administrador con permisos y Usuario sin permiso de uso del módulo “Usuarios”.

Si se desea actualizar la información de un usuario, hay que dar *click* en el icono “Editar” con la figura de un lápiz mostrado en la Figura 3.48.

Nombres	Apellidos	Nombre de usuario	Correo	Telefono	Admin	Actualizar	Eliminar
Verónica	López	verito	verito@hotmail.com	934271823	<input checked="" type="checkbox"/>		

Figura 3.48 Acciones Editar y Eliminar.

Después de presionar el icono “Editar”, se ingresa el valor a modificarse y se pulsa el botón “Guardar”. Finalmente, en la Figura 3.49 se comprueba el cumplimiento del RF32, puesto que se observa un mensaje de información indicando que la actualización ha sido exitosa.

The image shows a web form for user registration. At the top, there are two tabs: "Usuarios Registrados" and "Registrar Usuario", with the latter being active. The form fields are as follows:

- Nombres:** Verónica
- Apellidos:** López
- Nombre de usuario:** verito
- Email:** verito@gmail.com
- Contraseña:** (masked with dots)
- Telefono:** 934271823
- isAdmin:** A checkbox that is checked.

At the bottom of the form are two buttons: "Guardar" (with a save icon) and "Cancelar" (with a close icon). Overlaid on the right side of the form is a dark grey notification box with the text "Usuario actualizado" and a red "OK" button.

Figura 3.49 Mensaje de actualización exitosa de un usuario.

Por otra parte si se desea eliminar un usuario, se selecciona el icono “Eliminar” con la figura de un basurero mostrado previamente en la Figura 3.48. Después de pulsar el icono, aparece un cuadro de diálogo para confirmar la eliminación del usuario. El cumplimiento del RF33 se presenta en la Figura 3.50.

The image shows a confirmation dialog box. The text inside the dialog reads: "tesisaeie.centralus.cloudapp.azure.com dice" followed by "Esta seguro de eliminar este item?". At the bottom right of the dialog are two buttons: "Aceptar" (highlighted in blue) and "Cancelar". Below the dialog box is a dark grey notification box with the text "Usuario eliminado" and a red "OK" button.

Figura 3.50 Mensaje de eliminación exitosa para un usuario.

En la Figura 3.51 se observa una pestaña con el nombre “Usuarios Registrados”. En dicha pestaña se tiene una lista de los usuarios creados que podrán acceder al sistema. Además, en la lista se observa que los usuarios con la propiedad “Admin” activada poseen un *checkbox* activado a diferencia de los otros usuarios comunes que no la tienen.

Usuarios Registrados Registrar Usuario

USCAR

Nombre	Apellidos	Nombre de usuario	Correo	Telefono	Admin	Actualizar	Eliminar
admin	admin	administrator	admin@gmail.com	12312312	<input checked="" type="checkbox"/>		
Verónica	López	verito	verito@hotmail.com	934271823	<input checked="" type="checkbox"/>		
Alexandra	Paucar	alexita	alexita@gmail.com	923232323	<input type="checkbox"/>		

Items per page: 5 1 - 3 of 3 << < > >>

Figura 3.51 Lista de usuarios.

3.2.7 PRUEBAS DE VALIDACIÓN DEL MÓDULO MÓVIL

Este módulo corresponde a la aplicación móvil, y cumple la función de ser una interfaz informativa a los estudiantes. La Tabla 3.7 muestra los RF por cumplir en este módulo.

Tabla 3.7 Requerimientos Funcionales del Módulo Móvil.

CASO DE USO	RF	DESCRIPCION
CU07	RF35	Visualizar productos ofertados por la AEIE.
CU07	RF36	Visualizar cursos ofertados por la AEIE.
CU07	RF37	Visualizar eventos, notificaciones de tipo informativo publicados por la AEIE.
CU07	RF38	Visualizar la disponibilidad de casilleros.

La Figura 3.52 da cumplimiento al RF35 y RF36.



Figura 3.52 Interfaz módulo Móvil correspondiente a los menú Productos y Casilleros.

La Figura 3.53 da cumplimiento al RF37 y RF38.



Figura 3.53 Interfaz módulo Móvil correspondiente a los menú Cursos.

La Figura 3.54 da cumplimiento al RF38.



Figura 3.54 Interfaz módulo Móvil correspondiente a los menú Anuncios.

Concluidas las pruebas realizadas para cada módulo que compone el prototipo de software distribuido y obteniendo resultados satisfactorios, se puede verificar que el prototipo de gestión de productos y servicios ofertados por la AEIE cumple con las especificaciones y objetivo para el cual fue desarrollado. En la Tabla 3.8 y 3.9 se muestra la lista de los RF y el cumplimiento de los mismos.

Tabla 3.8 Lista de cumplimiento de los RF.

MÓDULOS	RF	DESCRIPCIÓN	CUMPLE
Catálogo	RF01	Añadir categoría de productos.	SI
	RF02	Modificar categoría de productos.	SI
	RF03	Eliminar categoría de productos.	SI
	RF04	Añadir productos.	SI
	RF05	Modificar productos.	SI
	RF06	Eliminar productos.	SI
	RF07	Añadir proveedor de productos.	SI
	RF08	Modificar proveedor productos.	SI
	RF09	Eliminar proveedor de productos.	SI
Venta	RF10	Registrar ventas del día.	SI
	RF11	Listar todas las ventas.	SI
Servicios	RF12	Registrar semestre.	SI
	RF13	Añadir estudiantes.	SI
	RF14	Modificar estudiantes.	SI
	RF15	Eliminar estudiantes.	SI
	RF16	Registrar el aporte de un estudiante.	SI
	RF17	Eliminar el aporte de un estudiante.	SI
	RF18	Añadir casilleros.	SI
	RF19	Modificar casilleros.	SI
	RF20	Eliminar casilleros.	SI
	RF21	Añadir el alquiler de casillero por un estudiante.	SI
	RF22	Eliminar el registro de un estudiante que alquila un casillero.	SI

Tabla 3.9 Lista de cumplimiento de los RF (continuación).

MÓDULOS	RF	DESCRIPCIÓN	CUMPLE
Servicios	RF23	Notificar mediante correo electrónico el estado de alquiler del casillero al estudiante.	SI
Cursos	RF24	Añadir cursos ofertados por la AEIE.	SI
	RF25	Modificar cursos ofertados por la AEIE.	SI
	RF26	Eliminar cursos ofertados por la AEIE.	SI
	RF27	Registrar estudiantes en los cursos.	SI
Informativo	RF28	Publicar eventos, avisos de tipo informativo que serán visualizados en la aplicación móvil destinada a los estudiantes.	SI
	RF29	Modificar avisos publicados.	SI
	RF30	Eliminar avisos publicados.	SI
Usuarios	RF31	Registrar usuario.	SI
	RF32	Modificar usuario.	SI
	RF33	Eliminar usuario.	SI
	RF34	Listar usuarios.	SI
Móvil	RF35	Visualizar productos ofertados por la AEIE.	SI
	RF36	Visualizar cursos ofertados por la AEIE.	SI
	RF37	Visualizar eventos, notificaciones de tipo informativo publicados por la AEIE.	SI
	RF38	Visualizar la disponibilidad de casilleros.	SI

3.3 PRUEBAS DE VALIDACIÓN DE REQUERIMIENTOS NO FUNCIONALES

En este apartado se verificará los RNF para el prototipo, descritos en el **Apartado 2.2.2.3**. En la Tabla 3.10, se detalla los RNF para los cuales se verificará su cumplimiento.

Tabla 3.10 Requerimientos No Funcionales.

	Requerimiento No Funcional	Descripción
Usabilidad	RNF01	El sistema debe brindar a los usuarios una interfaz gráfica amigable y de uso sencillo.
Escalabilidad	RNF02	El prototipo debe ser construido en una infraestructura que permita el incremento de recursos informáticos como memoria RAM y almacenamiento.
Disponibilidad	RNF03	El sistema debe estar disponible las 24 horas del día.
Autenticación	RNF04	El acceso al sistema web debe estar restringido por un correo electrónico y contraseña para cada usuario.

3.3.1 RNF01: USABILIDAD

Para la aplicación móvil se realizó una encuesta a 10 voluntarios entre estudiantes y egresados de las distintas carreras de la Facultad de Ingeniería Eléctrica y Electrónica; mientras que para la aplicación web se realizó la encuesta al personal de la AEIE (2 personas). Las encuestas se encuentran en el **ANEXO G** con un análisis de los resultados y fueron diseñadas mediante *Google Forms* [45].

En la Figura 3.55 se muestra el resultado de la pregunta 1 de la encuesta de satisfacción de la aplicación móvil. De los 10 voluntarios que respondieron a la encuesta el 100% respondió con una respuesta "SI". Por otra parte, en la Figura 3.56 se presenta el resultado de la pregunta 1 de la encuesta de satisfacción de la aplicación web. De las dos encuestas perteneciente al personal de la AEIE el 100% respondió con una respuesta "SI". En base a los resultados obtenidos de las encuestas realizadas se verifica el cumplimiento del RNF01 correspondiente a "Usabilidad".

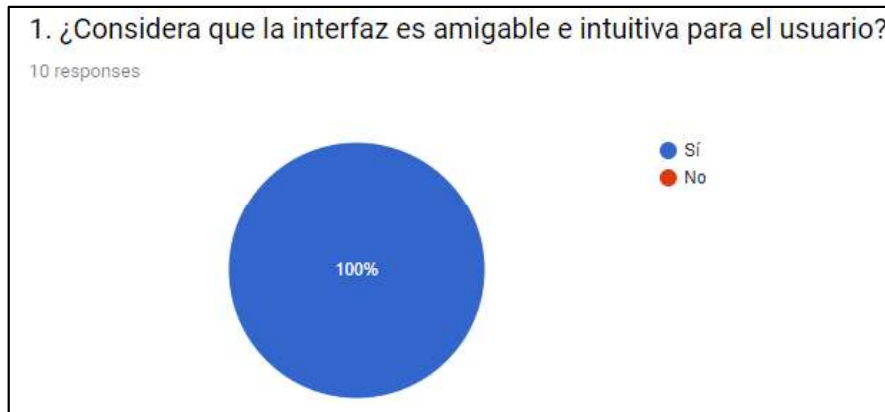


Figura 3.55 Respuesta de la pregunta 1 en la encuesta de satisfacción de la aplicación móvil.



Figura 3.56 Respuesta de la pregunta 1 en la encuesta de satisfacción de la aplicación web.

3.3.2 RN02: ESCALABILIDAD

La implementación del prototipo se llevó a cabo en el proveedor cloud “Microsoft Azure”. La escalabilidad es el proceso de administración de los recursos para permitir que una aplicación cumpla con un conjunto de requisitos de rendimiento. En Microsoft Azure el proceso de escalabilidad es una ventaja clave, ya que es una característica integrada en distintos servicios, tales como: *Cloud Services*, *Mobile Services*, *Virtual Machines* y *Websites*, entre otros [46]. Por ejemplo, el escalado vertical es el proceso de aumento de la capacidad de una instancia determinada. Una máquina virtual podría aumentarse de 1 vCPU y 3,5 GB de RAM a 2 vCPU y 7 GB de RAM para proporcionar más capacidad de procesamiento de manera sencilla [47]. Gracias a la facilidad de escalabilidad que ofrece

Microsoft Azure a sus clientes se da por cumplido el RNF02 correspondiente a “Escalabilidad”.

3.3.3 RNF03: DISPONIBILIDAD

La disponibilidad se enfoca principalmente en ofrecer un servicio o información que se encuentre accesible en todo momento para los distintos usuarios [48]. El contrato de nivel de servicios en Microsoft Azure (SLA) establece que: “*En el caso de cualquier Virtual Machine de Instancia Única con almacenamiento premium para todos los Discos del Sistema Operativo y Discos de Datos, garantizamos que dispondrá de Conectividad de Virtual Machine durante el 99,9 % del tiempo, como mínimo*” [49]. Por consiguiente, se verifica el cumplimiento del RNF03 correspondiente a “Disponibilidad”.

3.3.4 RNF04: AUTENTICACIÓN

El acceso al sistema web que gestionará la AEIE está restringido mediante un correo electrónico y contraseña para cada usuario. El usuario que desee iniciar sesión deberá ser previamente creado por un “Administrador”. En la Figura 3.55, se observa la interfaz mostrada al usuario junto con algunas validaciones como: “*Ingreso de email valido*” y “*Campo requerido*”.

The image shows a login interface with the following elements:

- Header:** The word "LOGIN" in large, bold, black capital letters.
- Email Field:** A text input box with a red border. The label "Email *" is in red. The text "verito" is entered. A red error message "Ingrese un email valido" is displayed below the field.
- Password Field:** A text input box with a red border. The label "Password *" is in red. A red error message "Campo requerido" is displayed below the field.
- Button:** A grey button labeled "Login" is positioned below the password field.

Figura 3.57 Interfaz de inicio de sesión.

Una vez que el usuario ha llenado los campos correctamente y ha pulsado el botón “*Login*”, ingresará a la página principal (*dashboard*) del sistema y visualizará su nombre de usuario como se muestra en la Figura 3.56. Caso contrario, se presentará el mensaje mostrado en la Figura 3.57 indicando “Autenticación fallida”. Por lo tanto, se da por cumplido el RNF04 correspondiente a “Autenticación”.



Figura 3.58 Nombre de usuario en la página principal.



Figura 3.59 Mensaje de autenticación fallida.

Habiendo terminado la validación de los RNF, a continuación se presenta en la Tabla 3.11 indicando el cumplimiento de los mismos.

Tabla 3.11 Lista de cumplimiento de los RNF.

	REQUERIMIENTO NO FUNCIONAL	DESCRIPCIÓN	CUMPLE
Usabilidad	RNF01	El sistema debe brindar a los usuarios una interfaz gráfica amigable y de uso sencillo.	SI
Escalabilidad	RNF02	El prototipo debe ser construido en una infraestructura que permita el incremento de recursos informáticos como memoria RAM y almacenamiento.	SI
Disponibilidad	RNF03	El sistema debe estar disponible las 24 horas del día.	SI
Autenticación	RNF04	El acceso al sistema web debe estar restringido por un correo electrónico y contraseña para cada usuario.	SI

3.4 TABLERO KANBAN FINAL

Finalizadas todas las etapas de Diseño, Implementación y Pruebas de este Proyecto Técnico se presenta el Tablero Final Kanban en la Figura 3.60.



Figura 3.60 Tablero Final Kanban.

4 CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

- Concluido este Proyecto Técnico se ha logrado conseguir el objetivo general de desarrollar un prototipo de software distribuido para la gestión de productos y servicios ofertados por la Asociación de Estudiantes de Ingeniería Electrónica (AEIE). El software distribuido está compuesto de una aplicación web y móvil, los mismos que permitirán el desempeño de las actividades de la AEIE de una manera más óptima; previniendo la pérdida de documentos y ahorrando espacio físico, así como también automatizando tareas realizadas manualmente.
- En la etapa de Diseño se ha tomado los requerimientos funcionales y no funcionales mediante entrevistas al personal de la AEIE. Además, con base en estos requerimientos, se procede a realizar los diagramas de casos de uso, diagrama de colecciones y diagrama de clases. Las acciones mencionadas permitieron diseñar los distintos módulos que conforman el prototipo de software distribuido con el fin de satisfacer las necesidades de la AEIE.
- El uso de una metodología ágil es importante, ya que permite conocer el estado del proyecto en cualquier momento. En este Proyecto Técnico fue elegida la metodología Kanban y aplicada a las distintas etapas que conllevaron a la consecución del mismo. Esta metodología posibilitó gestionar las distintas tareas gracias a la visualización del trabajo por fases, evitando la sobrecarga.
- La utilización del lenguaje JavaScript para la codificación del prototipo ha sido un gran aporte, ya que gracias a su versatilidad se lo pudo aplicar en el lado del servidor, cliente y base de datos. Este lenguaje de programación se encuentra presente en herramientas como MongoDB, Express, Angular, Node.js, las mismas que fueron estudiadas previamente con el fin de asociarlas y desarrollar la aplicación web perteneciente al prototipo de sistema distribuido planteado.
- El prototipo propuesto implementa la arquitectura de 3 capas (Presentación, Lógica de Negocio, Datos) que permite la distribución de la funcionalidad de la aplicación entre tres sistemas independientes. Estas capas están hospedadas en Microsoft Azure un proveedor Cloud Computing que ofrece servicios de cómputo para aprovisionar el servidor de manera sencilla y a través de una interfaz gráfica intuitiva y amigable.

- Los principales proveedores de Cloud Computing como Google Cloud, Amazon Web Services, Microsoft Azure poseen variados y excelentes servicios en la nube para la implementación de aplicaciones de software. Estos servicios pueden ser probados gratuitamente por 12 meses siempre y cuando se provea de una tarjeta de crédito al momento de registrar una cuenta nueva, esta acción se la realiza a modo de prevención en caso de que un usuario sobrepase los límites gratuitos. Sin embargo, esta estrategia resulta ser una limitante para estudiantes y jóvenes que no poseen una tarjeta de crédito. Por lo tanto, se escogió Microsoft Azure quien mantiene un convenio con la Escuela Politécnica Nacional para obtener 12 meses gratuitos de los servicios en la nube de Microsoft Azure a través del programa “*Azure for Students*” y sin el cual la implementación de este Proyecto Técnico no habría sido posible. Es importante mencionar que gracias a este convenio no se necesita de una tarjeta de crédito para crear la cuenta en Microsoft Azure.
- Los resultados obtenidos en la etapa de pruebas de los Requerimientos Funcionales y No Funcionales permitieron verificar el correcto comportamiento del prototipo. Además es importante mencionar que las pruebas son un proceso valioso para obedecer a la satisfacción del usuario final.

4.2. RECOMENDACIONES

- Es recomendable realizar el diseño de una aplicación de software pensando en la escalabilidad que pueda tener a futuro, pues dependiendo de esto realizar la escalabilidad se convertirá en algo sencillo o en un caos. Es así, que el desarrollo de software modular por capas es una alternativa para solventar dicho problema.
- Se recomienda escoger la metodología que mejor se adapte a los requerimientos del proyecto de software. La buena práctica del uso de una metodología permite al equipo de desarrollo avanzar de una forma organizada en la evolución del proyecto y garantizar los resultados en tiempos definidos.
- Se recomienda tomar el aplicativo móvil como punto de referencia para la investigación y desarrollo de una aplicación móvil complementaria para el sistema operativo iOS, con el fin de abarcar todos los usuarios posibles y no sólo usuarios con dispositivos Android.

- Se recomienda que la carrera de Ingeniería en Electrónica y Redes de Información profundice dentro de sus materias orientadas a programación temas como fundamentos y buenas prácticas de lenguajes de programación que permitan al estudiante estar en la capacidad de desarrollar aplicaciones web y móviles acorde al mercado laboral.

5 REFERENCIAS BIBLIOGRÁFICAS

- [1] B. Akiwate, A. Patel, T. Nabiwale, N. K. Naik, y S. Patil, "Web Based Student Information Management System using MEAN Stack", presentado en International Journal of Advanced Research in Computer Science and Software Engineering, 2016.
- [2] J. Organista-Sandoval, L. McAnally-Salas, y G. Lavigne, "El teléfono inteligente (smartphone) como herramienta pedagógica", *Apertura*, vol. 5, n.º 1, 2013.
- [3] Y. Jung, "What a smartphone is to me: understanding user values in using smartphones", *Information Systems Journal*, vol. 24, n.º 4, pp. 299-321, 2014.
- [4] "Gartner Says Worldwide Sales of Smartphones Returned to Growth in First Quarter of 2018", *Gartner*. [En línea]. Disponible en: <https://www.gartner.com/en/newsroom/press-releases/2018-05-29-gartner-says-worldwide-sales-of-smartphones-returned-to-growth-in-first-quarter-of-2018>. [Accedido: 07-mar-2019].
- [5] R. Meier, *Professional Android 4 Application Development*. John Wiley & Sons, 2012.
- [6] P. Mell y T. Grance, "The NIST Definition of Cloud Computing", p. 7.
- [7] M. Armbrust *et al.*, "A view of cloud computing", *Communications of the ACM*, vol. 53, n.º 4, p. 50, abr. 2010.
- [8] K. Jamsa, *Cloud Computing*. Jones & Bartlett Publishers, 2011.
- [9] M. D. Dikaiakos, D. Katsaros, P. Mehra, G. Pallis, y A. Vakali, "Cloud Computing: Distributed Internet Computing for IT and Scientific Research", *IEEE Internet Computing*, vol. 13, n.º 5, pp. 10-13, sep. 2009.
- [10] "Qué es Azure: Servicios en la nube de Microsoft | Microsoft Azure". [En línea]. Disponible en: <https://azure.microsoft.com/es-es/overview/what-is-azure/>. [Accedido: 17-mar-2019].
- [11] "What is AWS? - Amazon Web Services", *Amazon Web Services, Inc.* [En línea]. Disponible en: <https://aws.amazon.com/what-is-aws/>. [Accedido: 18-mar-2019].
- [12] "Google Cloud Platform Overview | Overview", *Google Cloud*. [En línea]. Disponible en: <https://cloud.google.com/docs/overview/>. [Accedido: 18-mar-2019].

- [13] K. Banker, *MongoDB in Action*. Greenwich, CT, USA: Manning Publications Co., 2011.
- [14] A. Silberschatz y H. Forth, *Fundamentos de Bases de datos*, 4ta ed. McGraw-Hill, 2014.
- [15] "What Is MongoDB?", *MongoDB*. [En línea]. Disponible en: <https://www.mongodb.com/what-is-mongodb>. [Accedido: 19-mar-2019].
- [16] G. Zhao, W. Huang, S. Liang, y Y. Tang, "Modeling MongoDB with Relational Model", en *2013 Fourth International Conference on Emerging Intelligent Data and Web Technologies*, 2013, pp. 115-121.
- [17] A. Mardan, "Boosting Node.js and MongoDB with Mongoose", en *Practical Node.js: Building Real-World Scalable Web Apps*, A. Mardan, Ed. Berkeley, CA: Apress, 2018, pp. 239-276.
- [18] S. Holmes, *Mongoose for Application Development*. Packt Publishing Ltd, 2013.
- [19] S. Chen, Z. Mao, Y.-M. Wang, y M. Zhang, "Pretty-Bad-Proxy: An Overlooked Adversary in Browsers' HTTPS Deployments", en *2009 30th IEEE Symposium on Security and Privacy*, Oakland, CA, USA, 2009, pp. 347-359.
- [20] S. Dustdar y W. Schreiner, "A survey on web services composition", *International Journal of Web and Grid Services*, vol. 1, n.º 1, p. 1, 2005.
- [21] M. Masse, *REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces*. O'Reilly Media, Inc., 2011.
- [22] A. Rodriguez, "RESTful Web services: The basics", *The basics*, p. 11.
- [23] D. Gaynes, *Node.js for .NET Developers*. Microsoft Press, 2015.
- [24] P. Teixeira, *Instant Node.js starter: program your scalable network applications and web services with Node.js*. Birmingham Mumbai: Packt Publishing, 2013.
- [25] "Pricing and Plans", *SendGrid*. [En línea]. Disponible en: <https://sendgrid.com/pricing/>. [Accedido: 23-sep-2019].
- [26] "#1 Push Service | Send Mobile & Web Push Notifications", *OneSignal*. [En línea]. Disponible en: <https://onesignal.com/>. [Accedido: 18-nov-2019].
- [27] "Angular - Architecture overview". [En línea]. Disponible en: <https://angular.io/guide/architecture>. [Accedido: 23-mar-2019].

- [28] “Documentation for Visual Studio Code”. [En línea]. Disponible en: <https://code.visualstudio.com/docs>. [Accedido: 22-mar-2019].
- [29] “Conoce Android Studio”, *Android Developers*. [En línea]. Disponible en: <https://developer.android.com/studio/intro>. [Accedido: 25-mar-2019].
- [30] E. Brechner, *Agile Project Management with Kanban*. Microsoft Press, 2015.
- [31] H. Kniberg y M. Skarin, *Kanban and Scrum - Making the Most of Both*. Lulu.com, 2010.
- [32] “Trello”. [En línea]. Disponible en: <https://trello.com>. [Accedido: 24-sep-2019].
- [33] “Gartner: Infraestructura de nube como servicio”, *Google Cloud*. [En línea]. Disponible en: <https://cloud.google.com/gartner-cloud-infrastructure-as-a-service/?hl=es-419>. [Accedido: 20-ago-2019].
- [34] “6 características clave de Microsoft Azure”, 20-dic-2015. .
- [35] “Pricing - Windows Virtual Machines | Microsoft Azure”. [En línea]. Disponible en: <https://azure.microsoft.com/en-us/pricing/details/virtual-machines/windows/>. [Accedido: 02-oct-2019].
- [36] “DbSchema: The Best Database Client & Diagram Designer”. [En línea]. Disponible en: <https://dbschema.com/>. [Accedido: 14-nov-2019].
- [37] P. C. G. de Marina, M. Á. G. Blázquez, y A. S. Alonso, *Especificando software mediante casos de USO y UML: Ejercicios resueltos*. Editorial Centro de Estudios Ramon Areces SA, 2019.
- [38] G. Bustos, “INTEGRACIÓN INFORMAL DE MODELOS EN UML”, p. 10.
- [39] T. Groß, “draw.io - Diagrams For Everyone, Everywhere”, *draw.io*. .
- [40] “NinjaMock online wireframe and mockup tool”, *NinjaMock*. [En línea]. Disponible en: <https://ninjamock.com>. [Accedido: 04-oct-2019].
- [41] “Download Visual Studio Code - Mac, Linux, Windows”. [En línea]. Disponible en: <https://code.visualstudio.com/Download>. [Accedido: 19-sep-2019].
- [42] “Download Android Studio and SDK tools”, *Android Developers*. [En línea]. Disponible en: <https://developer.android.com/studio>. [Accedido: 21-sep-2019].

- [43] “Consejos y Buenas Prácticas en Programación”. [En línea]. Disponible en: <https://personales.unican.es/sanchezbp/teaching/faqs/programming.html>. [Accedido: 06-oct-2019].
- [44] H. F. Nielsen, T. Berners-Lee, y R. T. Fielding, “Hypertext Transfer Protocol -- HTTP/1.0”. [En línea]. Disponible en: <https://tools.ietf.org/html/rfc1945>. [Accedido: 18-nov-2019].
- [45] “Formularios de Google: crea y analiza encuestas de forma gratuita”. [En línea]. Disponible en: <https://www.google.com/intl/es/forms/about/>. [Accedido: 18-nov-2019].
- [46] “Escalado automático de Azure | Microsoft Azure”. [En línea]. Disponible en: <https://azure.microsoft.com/es-es/features/autoscale/>. [Accedido: 04-nov-2019].
- [47] jboeshart, “Aprovechar el escalado vertical y horizontal de la arquitectura - Learn”. [En línea]. Disponible en: <https://docs.microsoft.com/es-es/learn/modules/design-for-performance-and-scalability-in-azure/2-scaling-up-and-scaling-out>. [Accedido: 04-nov-2019].
- [48] J. C. Santos, *Seguridad Informática (GRADO MEDIO)*. Grupo Editorial RA-MA.
- [49] “Contrato de nivel de servicio para Máquinas virtuales”. [En línea]. Disponible en: https://azure.microsoft.com/es-es/support/legal/sla/virtual-machines/v1_8/. [Accedido: 04-nov-2019].

ANEXOS

ANEXO A. Preguntas de las encuestas para toma de RF y RNF.

ANEXO B. Colecciones detalladas de la base de datos.

ANEXO C. Clases detalladas del diagrama de clases.

ANEXO D. Código del servicio web

ANEXO E. Código de la aplicación web desarrollada con Angular.

ANEXO F. Código de la aplicación Android.

ANEXO G. Encuestas de satisfacción de la aplicación web y móvil.

NOTA: Los Anexos se encuentran adjuntos en el CD.

ANEXO A

Preguntas de las encuestas respondidas por el personal de la AEIE para toma de RF y RNF.

ENCUESTA DE ESTABLECIMIENTO DEL PROBLEMA

¿Ha tenido problemas de pérdida e inconsistencia de datos?

- Sí

¿Los datos almacenados en la computadora de la AEIE tienen respaldos?

- No todos.

¿Existen tareas que se realicen de forma manual?

- Ventas de productos, notificación del desalojo de casillero, registro de cursos.

¿Ha tenido problemas de pérdida e inconsistencia de datos en archivos de Excel?

- Sí

¿Qué tareas se registran en archivos de excel?

- Registro de aportantes, alquiler de casilleros, registro de cursos.

¿Está de acuerdo con que se desarrolle un software que permita gestionar a la AEIE sus productos y servicios ofertados?

- Sí

¿Consideraría utilizar un prototipo de sistema distribuido que permita desempeñar sus actividades de una mejor manera, estando en capacidad de acceder a este desde cualquier computador y evitando la inconsistencia de datos?

- Sí

ENCUESTA DEL MÓDULO CATÁLOGO

Este módulo le permitiría añadir, modificar, eliminar y visualizar los productos ofertados de la AEIE. ¿Está de acuerdo con su implementación?

- Sí

¿Qué le parece si se agrega la opción de tener categorías y proveedores para los productos para complementar la funcionalidad de este módulo?

- Es una buena idea.

ENCUESTA DEL MÓDULO VENTAS

¿Cuál es el método que usa para registrar las ventas?

- Es un método manual donde se registra el producto, su precio y la cantidad vendida en un cuaderno.

Este módulo le permitiría realizar la venta de los productos ofertados de la AEIE.

¿Está de acuerdo con su implementación?

- Sí

¿Estaría de acuerdo con la implementación de una interfaz amigable donde se seleccione los productos de la AEIE para posteriormente agregarlos a una lista de productos para guardar la venta?

- Es una buena idea.

¿Qué le parece si se implementa y muestra la lista de ventas con el usuario que las realizó?

- Estoy de acuerdo

ENCUESTA DEL MÓDULO SERVICIOS

Este módulo le permitiría gestionar la información de un semestre referencial; gestionar estudiantes, aportes y casilleros. ¿Está de acuerdo con su implementación?

- Sí

¿Estaría de acuerdo con la implementación de la gestión de estudiantes. Las acciones permitidas serían añadir, eliminar, modificar la información de estudiantes?

- Estoy de acuerdo

¿Estaría de acuerdo con la implementación de la gestión de aporte. Las acciones permitidas serían añadir, eliminar, modificar la información de aportes?

- Estoy de acuerdo

¿Estaría de acuerdo con la implementación de la gestión de aporte. Las acciones permitidas serían añadir, eliminar, modificar la información de los casilleros?

- Estoy de acuerdo

¿Qué le parece si se implementa un método que envíe automáticamente un correo indicando el estado del alquiler de un casillero al estudiante para evitar problemas de desconocimiento?

- Estoy de acuerdo

¿Qué le parece si se implementa una interfaz amigable que le permita seleccionar un casillero y posteriormente asignarlo a un estudiante desde una lista de estudiantes?

- Estoy de acuerdo con la implementación

ENCUESTA DEL MÓDULO CASILLEROS

Este módulo le permitiría gestionar los casilleros de la AEIE, con este módulo estaría en capacidad de añadir, modificar o eliminar casilleros. ¿Está de acuerdo con su implementación?

- Sí

ENCUESTA DEL MÓDULO USUARIOS

Este módulo le permitiría gestionar los casilleros de la AEIE, con este módulo estaría en capacidad de añadir, modificar o eliminar usuarios. Además otorgaría seguridad, ya que solo los usuarios registrados podrán acceder a la aplicación web. ¿Está de acuerdo con su implementación?

- Sí

¿Estaría de acuerdo con la implementación de un rol administrador que permite gestionar a los usuarios que podrán iniciar sesión en la aplicación web para brindar seguridad y autenticación al prototipo propuesto?

- Sí

ENCUESTA DEL MÓDULO INFORMATIVO

¿Existen estudiantes que no tienen una cuenta en la red social Facebook donde la AEIE publica avisos de tipo informativo?

- Sí

Este módulo le permitiría publicar avisos de tipo informativo que serán visualizados por los estudiantes mediante una aplicación móvil favoreciendo a la interacción entre ambas partes. ¿Está de acuerdo con su implementación?

- De acuerdo

Este módulo le permitiría gestionar los avisos de tipo informativo con acciones como: añadir, modificar, eliminar avisos. ¿Está de acuerdo con su implementación?

- Estoy de acuerdo.

ENCUESTA DEL MÓDULO MÓVIL

Este módulo le permitiría a un estudiante visualizar los productos, cursos, casilleros, y anuncios de la AEIE mediante una aplicación móvil con una interfaz de tipo informativo. Además, favorecería a la interacción entre ambas partes ¿Está de acuerdo con su implementación?

- De acuerdo

ENCUESTA DE REQUERIMIENTOS NO FUNCIONALES

¿Qué otros aspectos espera del software propuesto?

- Una interfaz amigable

¿Consideraría positivo que el software se encuentre en la nube evitando pérdida de datos en su computador local y esté disponible 24 horas del día para que pueda cumplir con sus actividades?

- Sería una gran ventaja

¿Qué le parece si se restringe el acceso a la aplicación web mediante el inicio de sesión mediante email y contraseña?

- Sería buena idea ya que añadiría seguridad.

ANEXO B

Colecciones detalladas de la base de datos.

Colección Proveedor		
Campo	Tipo de dato	Descripción
_id	<i>String</i>	Identificador único del proveedor.
Nombre	<i>String</i>	Nombre del proveedor.
Teléfono	<i>String</i>	Teléfono del proveedor.
Estado	<i>Boolean</i>	Indicador para ocultar el proveedor cuando se elimine.

Colección Producto		
Campo	Tipo de dato	Descripción
_id	<i>String</i>	Identificador único de un producto.
Nombre	<i>String</i>	Nombre del proveedor.
Descripción	<i>String</i>	Información adicional que caracterice a un producto.
Precio	<i>Double</i>	Precio de un producto.
Stock	<i>Integer</i>	Cantidad de productos disponibles.
Categoría	<i>String</i>	Identificador único que se relaciona a una categoría.
Proveedor	<i>String</i>	Identificador único que se relaciona a un proveedor.
Estado	<i>Boolean</i>	Indicador para ocultar el producto cuando se elimine.

Colección Venta		
Campo	Tipo de dato	Descripción
_id	<i>String</i>	Identificador único de una venta
Fecha	<i>Date</i>	Fecha de la venta realizada.
Valor total	<i>Double</i>	Apellidos del usuario.
Productos	<i>Object</i>	Documento embebido con distintos productos.

Colección Semestre		
Campo	Tipo de dato	Descripción
_id	<i>String</i>	Identificador único de un semestre.
Nombre	<i>String</i>	Nombre referencial para el semestre.

Colección Usuario		
Campo	Tipo de dato	Descripción
_id	<i>String</i>	Identificador único de un usuario.
Nombres	<i>String</i>	Nombres del usuario.
Apellidos	<i>String</i>	Apellidos del usuario.
Username	<i>String</i>	Nombre de un usuario.
Email	<i>String</i>	Correo electrónico de un usuario.
Contraseña	<i>String</i>	Contraseña cifrada del usuario.
Teléfono	<i>String</i>	Teléfono del usuario
Admin	<i>Boolean</i>	Indicador para determinar si un usuario puede tener el rol de Administrador.
Estado	<i>Boolean</i>	Indicador para ocultar el usuario cuando se elimine.

Colección Curso		
Campo	Tipo de dato	Descripción
_id	<i>String</i>	Identificador único de un curso.
Nombres	<i>String</i>	Nombres del curso.
Descripción	<i>String</i>	Información adicional que caracteriza el curso.
Fecha	<i>Date</i>	Fecha de inicio del curso.
Precio	<i>Double</i>	Precio del curso.
Cupos	<i>Integer</i>	Cupos disponibles del el curso.
Semestre	<i>String</i>	Identificador referente a un semestre.
Estudiantes	<i>Object</i>	Documento embebido con varios estudiantes.
Estado	<i>Boolean</i>	Indicador para ocultar el curso cuando se elimine.

Colección Carrera		
Campo	Tipo de dato	Descripción
_id	<i>String</i>	Identificador único de una carrera.
Nombre	<i>String</i>	Nombre referencial para la carrera.

Colección Estudiante		
Campo	Tipo de dato	Descripción
_id	<i>String</i>	Identificador único de un estudiante.
Nombres	<i>String</i>	Nombres del estudiante.
Apellidos	<i>String</i>	Apellidos del estudiante.
Carrera	<i>String</i>	Identificador referente a una Carrera.
Cédula	<i>String</i>	Cédula del estudiante.
Número único	<i>String</i>	Número único del estudiante.
Email	<i>String</i>	Correo electrónico del estudiante.
Teléfono	<i>String</i>	Teléfono del estudiante
Estado	<i>Boolean</i>	Indicador para ocultar el estudiante cuando se elimine.

Colección Aporte		
Campo	Tipo de dato	Descripción
_id	<i>String</i>	Identificador único de un aporte.
Fecha	<i>Date</i>	Fecha de pago del aporte.
Semestre	<i>String</i>	Identificador que referencia al semestre.
Estudiante	<i>String</i>	Identificador del estudiante que realiza el aporte.

Colección Casilleros		
Campo	Tipo de dato	Descripción
_id	<i>String</i>	Identificador de un casillero.
Descripción	<i>String</i>	Información adicional.
Libre	<i>Boolean</i>	Indica si está ocupado o libre el casillero.
Estado	<i>Boolean</i>	Indicador para ocultar el casillero cuando se elimine.

Colección Ads		
Campo	Tipo de dato	Descripción
_id	<i>String</i>	Identificador único de un anuncio.
Título	<i>String</i>	Título de un anuncio.
Descripción	<i>String</i>	Descripción de un anuncio.

ANEXO C

Clases detalladas del diagrama de clases.

Proveedor		
Atributo	Tipo de dato	Descripción
_id	<i>String</i>	Identificador único del proveedor.
Nombre	<i>String</i>	Nombre del proveedor.
Teléfono	<i>String</i>	Teléfono del proveedor.
Estado	<i>Boolean</i>	Indicador para ocultar el proveedor cuando se elimine.

Producto		
Atributo	Tipo de dato	Descripción
_id	<i>String</i>	Identificador único de un producto.
Nombre	<i>String</i>	Nombre del proveedor.
Descripción	<i>String</i>	Información adicional que caracterice a un producto.
Precio	<i>Double</i>	Precio de un producto.
Stock	<i>Integer</i>	Cantidad de productos disponibles.
Categoría	<i>String</i>	Identificador único que se relaciona a una categoría.
Proveedor	<i>String</i>	Identificador único que se relaciona a un proveedor.
Estado	<i>Boolean</i>	Indicador para ocultar el producto cuando se elimine.

Venta		
Atributo	Tipo de dato	Descripción
_id	<i>String</i>	Identificador único de una venta
Fecha	<i>Date</i>	Fecha de la venta realizada.
Valor total	<i>Double</i>	Apellidos del usuario.
Productos	<i>Object</i>	Documento embebido con distintos productos.

Semestre		
Atributo	Tipo de dato	Descripción
_id	<i>String</i>	Identificador único de un semestre.
Nombre	<i>String</i>	Nombre referencial para el semestre.

Usuario		
Atributo	Tipo de dato	Descripción
_id	<i>String</i>	Identificador único de un usuario.
Nombres	<i>String</i>	Nombres del usuario.
Apellidos	<i>String</i>	Apellidos del usuario.
Username	<i>String</i>	Nombre de un usuario.
Email	<i>String</i>	Correo electrónico de un usuario.
Contraseña	<i>String</i>	Contraseña cifrada del usuario.
Teléfono	<i>String</i>	Teléfono del usuario
Admin	<i>Boolean</i>	Indicador para determinar si un usuario puede tener el rol de Administrador.
Estado	<i>Boolean</i>	Indicador para ocultar el usuario cuando se elimine.

Curso		
Atributo	Tipo de dato	Descripción
_id	<i>String</i>	Identificador único de un curso.
Nombres	<i>String</i>	Nombres del curso.
Descripción	<i>String</i>	Información adicional que caracteriza el curso.
Fecha	<i>Date</i>	Fecha de inicio del curso.
Precio	<i>Double</i>	Precio del curso.
Cupos	<i>Integer</i>	Cupos disponibles del el curso.
Semestre	<i>String</i>	Identificador referente a un semestre.
Estudiantes	<i>Object</i>	Documento embebido con varios estudiantes.
Estado	<i>Boolean</i>	Indicador para ocultar el curso cuando se elimine.

Carrera		
Atributo	Tipo de dato	Descripción
_id	<i>String</i>	Identificador único de una carrera.
Nombre	<i>String</i>	Nombre referencial para la carrera.

Estudiante		
Atributo	Tipo de dato	Descripción
_id	<i>String</i>	Identificador único de un estudiante.
Nombres	<i>String</i>	Nombres del estudiante.
Apellidos	<i>String</i>	Apellidos del estudiante.
Carrera	<i>String</i>	Identificador referente a una Carrera.
Cédula	<i>String</i>	Cédula del estudiante.
Número único	<i>String</i>	Número único del estudiante.
Email	<i>String</i>	Correo electrónico del estudiante.
Teléfono	<i>String</i>	Teléfono del estudiante
Estado	<i>Boolean</i>	Indicador para ocultar el estudiante cuando se elimine.

Aporte		
Atributo	Tipo de dato	Descripción
_id	<i>String</i>	Identificador único de un aporte.
Fecha	<i>Date</i>	Fecha de pago del aporte.
Semestre	<i>String</i>	Identificador que referencia al semestre.
Estudiante	<i>String</i>	Identificador del estudiante que realiza el aporte.

Casilleros		
Atributo	Tipo de dato	Descripción
_id	<i>String</i>	Identificador de un casillero.
Descripción	<i>String</i>	Información adicional.
Libre	<i>Boolean</i>	Indica si está ocupado o libre el casillero.
Estado	<i>Boolean</i>	Indicador para ocultar el casillero cuando se elimine.

Ads		
Atributo	Tipo de dato	Descripción
_id	<i>String</i>	Identificador único de un anuncio.
Título	<i>String</i>	Título de un anuncio.
Descripción	<i>String</i>	Descripción de un anuncio.

ANEXO G

Encuestas de satisfacción de la aplicación web y móvil. Modelos y Análisis de resultados.

Encuesta de Satisfacción de Aplicación Móvil

Esta encuesta permitirá analizar los requerimientos no funcionales del trabajo de titulación con el tema "DESARROLLO DE UN PROTOTIPO DE SOFTWARE DISTRIBUIDO PARA LA GESTIÓN DE PRODUCTOS Y SERVICIOS OFERTADOS POR LA ASOCIACIÓN DE ESTUDIANTES DE INGENIERÍA ELECTRÓNICA"

* Required

1. ¿Considera que la interfaz es amigable e intuitiva para el usuario? *

Sí

No

2. ¿Considera que el rendimiento de la aplicación en su smartphone fue lento o normal? *

Lento

Normal

3. ¿Tuvo algún problema al visualizar la información en los menús Producto, Casilleros, Cursos o Anuncios? En tal caso seleccione el o los menús correspondientes, caso contrario si no hubo ningún problema seleccione Ninguno. *

Productos

Casilleros

Cursos

Anuncios

Ninguno

4. ¿Qué marca es su smartphone? *

Your answer _____

5. ¿Cómo calificaría a la aplicación en un rango del 1 al 5? *

1 2 3 4 5

Mala Excelente

6. ¿Tiene algunas ideas sobre cómo mejorar este software?

Your answer _____

SUBMIT

Never submit passwords through Google Forms.

Encuesta de satisfacción de aplicación web

Esta encuesta permitirá analizar los requerimientos no funcionales del trabajo de titulación con el tema "DESARROLLO DE UN PROTOTIPO DE SOFTWARE DISTRIBUIDO PARA LA GESTIÓN DE PRODUCTOS Y SERVICIOS OFERTADOS POR LA ASOCIACIÓN DE ESTUDIANTES DE INGENIERÍA ELECTRÓNICA"

1. ¿Considera que la interfaz es amigable e intuitiva para el usuario? *

- Sí
- No

2. ¿Considera que el rendimiento de la aplicación web fue lento o normal? *

- Lento
- Normal

3. ¿La interacción con el módulo Catálogo fue? *

- Fácil
- Relativamente fácil
- Complicada
- Muy Difícil

4. ¿La interacción con el módulo Ventas fue? *

- Fácil
- Relativamente fácil
- Complicada
- Muy Difícil

Section title (optional)

Description (optional)

5. ¿La interacción con el módulo Servicios fue? *

- Fácil
- Relativamente fácil
- Complicada
- Muy Difícil

6. ¿La interacción con el módulo Cursos fue? *

- Fácil
- Relativamente fácil
- Complicada
- Muy Difícil

6. ¿La interacción con el módulo Informativo fue? *

- Fácil
- Relativamente fácil
- Complicada
- Muy Difícil

8. ¿Cómo calificaría a la aplicación en un rango del 1 al 5? *

	1	2	3	4	5	
Mala	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Excelente

Análisis de las respuestas perteneciente a las encuestas para la aplicación móvil

Timestamp	1. ¿Considera que la interfaz es amigable e intuitiva para el usuario?	2. ¿Considera que el rendimiento de la aplicación en su smartphone fue lento o normal?	3. ¿Tuvo algún problema al visualizar la información en los menús Producto, Casilleros, Cursos o Anuncios? En tal caso seleccione el o los menús correspondientes, caso contrario si no hubo ningún problema seleccione Ninguno.	4. ¿Qué marca es su smartphone?	5. ¿Cómo calificaría a la aplicación en un rango del 1 al 5?
11/4/2019 18:03:07	Sí	Normal	Ninguno	Xiaomi	4
11/4/2019 18:36:08	Sí	Normal	Ninguno	Huawei	4
11/4/2019 18:41:35	Sí	Normal	Ninguno	Huawei	4
11/4/2019 18:43:22	Sí	Normal	Ninguno	Xiaomi Pocophone F1	4
11/4/2019 18:48:25	Sí	Normal	Ninguno	Huawei	4
11/4/2019 18:53:37	Sí	Normal	Ninguno	Samsung	5
11/4/2019 19:13:54	Sí	Normal	Ninguno	Samsung Galaxy J2 Pro	5
11/4/2019 19:17:16	Sí	Normal	Ninguno	samsung S7	4
11/4/2019 19:57:37	Sí	Normal	Ninguno	Huawei	4
11/4/2019 23:23:38	Sí	Normal	Ninguno	Huawei	4

Análisis de las respuestas perteneciente a las encuestas para la aplicación web

Timestamp	1. ¿Considera que la interfaz es amigable e intuitiva para el usuario?	2. ¿Considera que el rendimiento de la aplicación web fue lento o normal?	3. ¿La interacción con el módulo Catálogo fue?	4. ¿La interacción con el módulo Ventas fue?	5. ¿La interacción con el módulo Servicios fue?	6. ¿La interacción con el módulo Cursos fue?	7. ¿La interacción con el módulo Informativo fue?	8. ¿Cómo calificaría a la aplicación en un rango del 1 al 5?
11/5/2019 22:53:11	Sí	Normal	Relativamente fácil	Fácil	Fácil	Fácil	Fácil	4
11/5/2019 22:54:12	Sí	Normal	Fácil	Fácil	Relativamente fácil	Fácil	Fácil	5

ORDEN DE EMPASTADO