

ESCUELA POLITÉCNICA NACIONAL

**FACULTAD DE INGENIERÍA ELÉCTRICA Y
ELECTRÓNICA**

**DESARROLLO DE UN PROTOTIPO DE UNA APLICACIÓN PARA
TELÉFONOS INTELIGENTES QUE PERMITA GESTIONAR EL
REGISTRO FARMACÉUTICO DE MEDICAMENTOS**

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

ANDRÉS MARCELO PARRA ACOSTA
andres.parra@epn.edu.ec

DIRECTOR: M.Sc. XAVIER ALEXANDER CALDERÓN HINOJOSA
xavier.calderon@epn.edu.ec

Quito, Julio 2020

AVAL

Certifico que el presente trabajo fue desarrollado por Andrés Marcelo Parra Acosta, bajo mi supervisión.

M.Sc. Xavier Alexander Calderón Hinojosa
DIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo, Andrés Marcelo Parra Acosta, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada 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 que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.

Andrés Marcelo Parra Acosta

DEDICATORIA

A mis padres y hermanos, quienes han estado a mi lado en cada momento de mi vida como pilar fundamental de apoyo.

A E. G., quien en vida fue una persona excepcional con muchos valores y principios, y siempre está presente en mi mente.

A P. P., quien partió de este mundo a muy corta edad y me dio mucha fortaleza para continuar a pesar de las dificultades.

AGRADECIMIENTO

A mis padres, por ser comprensivos e inculcarme valores para alcanzar las metas propuestas.

A N. y G., por formar parte de mi vida y apoyarme en todo momento, sin importar las circunstancias.

A mis amigos y compañeros, con quienes compartí muchas experiencias durante mi vida estudiantil universitaria.

ÍNDICE DE CONTENIDO

AVAL.....	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO	IV
ÍNDICE DE CONTENIDO.....	V
ÍNDICE DE FIGURAS	IX
ÍNDICE DE TABLAS.....	XIV
ÍNDICE DE CÓDIGOS	XVI
RESUMEN.....	XVII
ABSTRACT.....	XVIII
1 INTRODUCCIÓN	1
1.1 OBJETIVOS.....	2
1.2 ALCANCE	2
1.3 MARCO TEÓRICO	5
1.3.1 BASE DE DATOS.....	5
1.3.1.1 SISTEMA DE GESTIÓN DE BASES DE DATOS (DBMS).....	5
1.3.1.2 MODELO RELACIONAL	6
1.3.1.3 LENGUAJE DE CONSULTA ESTRUCTURADO (SQL).....	7
1.3.2 SERVICIO WEB	7
1.3.2.1 REST (REPRESENTATIONAL STATE TRANSFER).....	8
1.3.3 CLOUD COMPUTING	9
1.3.3.1 TIPOS DE CLOUD COMPUTING	10
1.3.3.2 TIPOS DE SERVICIOS	10
1.3.3.3 BENEFICIOS DE CLOUD COMPUTING	11
1.3.3.4 AMAZON WEB SERVICES	12
1.3.3.5 APLICACIONES MÓVILES	14
1.3.3.6 TIPOS DE APLICACIONES	14
1.3.3.7 ANDROID.....	16
1.3.3.8 RETROFIT	18
1.3.4 HERRAMIENTAS DE DESARROLLO	20
1.3.4.1 MYSQL.....	20
1.3.4.2 VISUAL STUDIO CODE	20
1.3.4.3 NODE.JS.....	20
1.3.4.4 ANDROID STUDIO.....	21

1.3.5	KANBAN	22
1.3.5.1	TABLERO KANBAN	23
1.3.5.2	TARJETAS KANBAN.....	23
1.3.5.3	BENEFICIOS DE KANBAN	23
2	METODOLOGÍA.....	24
2.1	DISEÑO	24
2.1.1	TABLERO KANBAN	25
2.1.2	ANÁLISIS DE REQUERIMIENTOS	26
2.1.2.1	RESULTADOS DE ENTREVISTA A ADMINISTRADOR	26
2.1.2.2	RESULTADOS DE ENTREVISTAS A MÉDICOS	28
2.1.2.3	RESULTADOS DE ENTREVISTAS A DEPENDIENTES DE FARMACIA	30
2.1.2.4	REQUERIMIENTOS FUNCIONALES	32
2.1.2.5	REQUERIMIENTOS NO FUNCIONALES.....	34
2.1.3	MÓDULOS DEL PROTOTIPO.....	34
2.1.4	ARQUITECTURA DEL PROTOTIPO.....	40
2.1.5	DISEÑO DE CAPA DE DATOS	40
2.1.5.1	DIAGRAMA RELACIONAL.....	40
2.1.6	DISEÑO DE CAPA DE NEGOCIO.....	43
2.1.6.1	DIAGRAMA DE CASOS DE USO	43
2.1.6.2	DIAGRAMA DE ACTIVIDADES	47
2.1.6.3	DIAGRAMA DE CLASES	56
2.1.7	DISEÑO DE CAPA DE PRESENTACIÓN	57
2.2	IMPLEMENTACIÓN.....	70
2.2.1	ACTUALIZACIÓN DEL TABLERO KANBAN.....	70
2.2.2	CONFIGURACIÓN DE AWS.....	71
2.2.3	CODIFICACIÓN.....	73
2.2.3.1	CODIFICACIÓN DE CAPA DE DATOS	73
2.2.3.2	CODIFICACIÓN DE CAPA DE NEGOCIO	75
2.2.3.3	CONECTIVIDAD LOCAL	81
2.2.3.4	MIGRACIÓN DE SERVICIOS A AWS	82
2.2.3.5	CONECTIVIDAD EN LA NUBE	83
2.2.3.6	CODIFICACIÓN DE CAPA DE PRESENTACIÓN.....	84
3	RESULTADOS Y DISCUSIÓN	91
3.1	ACTUALIZACIÓN DEL TABLERO KANBAN.....	91
3.2	PRUEBAS DE FUNCIONAMIENTO	92
3.2.1	PRUEBAS DE VALIDACIÓN DE REQUERIMIENTOS FUNCIONALES: MODULO REGISTRO DE USUARIO	92

3.2.1.1	REQUERIMIENTO FUNCIONAL RF01: CREAR NUEVO USUARIO.	92
3.2.2	PRUEBAS DE VALIDACIÓN DE REQUERIMIENTOS FUNCIONALES: MÓDULO INICIO DE SESIÓN DE USUARIO	98
3.2.2.1	REQUERIMIENTO FUNCIONAL RF02: INGRESAR CREDENCIALES DE USUARIO	98
3.2.3	PRUEBAS DE VALIDACIÓN DE REQUERIMIENTOS FUNCIONALES: MÓDULO DATOS PERSONALES DE USUARIO	102
3.2.3.1	REQUERIMIENTO FUNCIONAL RF03: VISUALIZAR DATOS DE USUARIO	102
3.2.3.2	REQUERIMIENTO FUNCIONAL RF04: ACTUALIZAR DATOS DE USUARIO	102
3.2.3.3	REQUERIMIENTO FUNCIONAL RF05: ACTUALIZAR CONTRASEÑA DE USUARIO	103
3.2.3.4	REQUERIMIENTO FUNCIONAL RF06: DESACTIVAR USUARIO ..	104
3.2.3.5	REQUERIMIENTO FUNCIONAL RF07: ELIMINAR USUARIO	105
3.2.4	PRUEBAS DE VALIDACIÓN DE REQUERIMIENTOS FUNCIONALES: MÓDULO NUEVO STOCK DE MEDICAMENTO	106
3.2.4.1	REQUERIMIENTO FUNCIONAL RF08: INGRESAR DATOS DE UN MEDICAMENTO.....	106
3.2.4.2	REQUERIMIENTO FUNCIONAL RF09: CREAR STOCK DE MEDICAMENTO.....	109
3.2.5	PRUEBAS DE VALIDACIÓN DE REQUERIMIENTOS FUNCIONALES: MÓDULO LISTA DE MEDICAMENTOS EN STOCK	111
3.2.5.1	REQUERIMIENTO FUNCIONAL RF10: VISUALIZAR MEDICAMENTOS EN STOCK	111
3.2.5.2	REQUERIMIENTO FUNCIONAL RF11: ELIMINAR MEDICAMENTO EN STOCK	112
3.2.6	PRUEBAS DE VALIDACIÓN DE REQUERIMIENTOS FUNCIONALES: MÓDULO NUEVA ATENCIÓN MÉDICA	114
3.2.6.1	REQUERIMIENTO FUNCIONAL RF16: INGRESAR DATOS DE PACIENTE	114
3.2.6.2	REQUERIMIENTO FUNCIONAL RF17: SELECCIONAR MEDICAMENTO DE RECETA.....	115
3.2.6.3	REQUERIMIENTO FUNCIONAL RF18: CREAR ATENCIÓN MÉDICA	116
3.2.7	PRUEBAS DE VALIDACIÓN DE REQUERIMIENTOS FUNCIONALES: MÓDULO ATENCIONES MÉDICAS POR ENTREGAR MEDICAMENTOS	118
3.2.7.1	REQUERIMIENTO FUNCIONAL RF22: BUSCAR PACIENTE REGISTRADO	118
3.2.7.2	REQUERIMIENTO FUNCIONAL RF23: VISUALIZAR ATENCIÓN MÉDICA PENDIENTE DE ENTREGA DE MEDICAMENTOS DE PACIENTE.....	119
3.2.7.3	REQUERIMIENTO FUNCIONAL RF24: REGISTRAR ENTREGA DE MEDICAMENTO DE RECETA.....	120

3.2.7.4	REQUERIMIENTO FUNCIONAL RF25: ACTUALIZAR ATENCIÓN MÉDICA	122
3.2.8	PRUEBAS DE VALIDACIÓN DE REQUERIMIENTOS FUNCIONALES: MÓDULO ATENCIONES MÉDICAS REGISTRADAS	125
3.2.8.1	REQUERIMIENTO FUNCIONAL RF19: VISUALIZAR ATENCIÓN MÉDICA REGISTRADA	125
3.2.8.2	REQUERIMIENTO FUNCIONAL RF20: REEMPLAZAR MEDICAMENTO DE RECETA POR OTRO	127
3.2.8.3	REQUERIMIENTO FUNCIONAL RF21: ELIMINAR MEDICAMENTO DE RECETA	129
3.2.9	PRUEBAS DE VALIDACIÓN DE REQUERIMIENTOS FUNCIONALES: MÓDULO ATENCIONES MÉDICAS FINALIZADAS	131
3.2.9.1	REQUERIMIENTO FUNCIONAL RF26: VISUALIZAR ATENCIÓN MÉDICA FINALIZADA	131
3.2.10	PRUEBAS DE VALIDACIÓN DE REQUERIMIENTOS FUNCIONALES: MÓDULO HISTORIAL FARMACÉUTICO DE MEDICAMENTO	132
3.2.10.1	REQUERIMIENTO FUNCIONAL RF12: SELECCIONAR UN MEDICAMENTO PARA HISTORIAL	132
3.2.10.2	REQUERIMIENTO FUNCIONAL RF13: ESCOGER TIPO DE HISTORIAL	133
3.2.10.3	REQUERIMIENTO FUNCIONAL RF14: GENERAR HISTORIAL DE MEDICAMENTO	133
3.2.10.4	REQUERIMIENTO FUNCIONAL RF15: CREAR PDF CON HISTORIAL DE MEDICAMENTO	134
3.3	RESULTADOS DE ENCUESTAS DE VALIDACIÓN DE FUNCIONAMIENTO	136
3.4	TABLERO KANBAN FINAL	140
4	CONCLUSIONES Y RECOMENDACIONES	142
4.1	CONCLUSIONES	142
4.2	RECOMENDACIONES	143
5	REFERENCIAS BIBLIOGRÁFICAS	145
6	ANEXOS	149
	ANEXO A. ENTREVISTAS PARA OBTENCIÓN DE REQUERIMIENTOS	149
	ANEXO B. CÓDIGO BASE DE DATOS	149
	ANEXO C. CÓDIGO SERVICIO WEB REST	149
	ANEXO D. CÓDIGO ANDROID STUDIO	149
	ANEXO E. ENCUESTAS DE VALIDACIÓN DE FUNCIONAMIENTO	149
	ORDEN DE EMPASTADO	150

ÍNDICE DE FIGURAS

Figura 1.1. Esquema de funcionamiento del Proyecto.....	4
Figura 1.2. Tabla de Base de Datos	6
Figura 1.3. Características de Instancia EC2	13
Figura 1.4. Características de Instancia RDS.....	14
Figura 1.5. Arquitectura de Android	18
Figura 2.1. Arquitectura del Prototipo	41
Figura 2.2. Diagrama Relacional.....	42
Figura 2.3. Diagrama de Casos de Uso: Nuevo Stock de Medicamento.....	45
Figura 2.4. Diagrama de Casos de Uso: Lista de Medicamentos en Stock.....	45
Figura 2.5. Diagrama de Casos de Uso: Historial Farmacéutico de Medicamento	45
Figura 2.6. Diagrama de Casos de Uso: Nueva Atención Médica	46
Figura 2.7. Diagrama de Casos de Uso: Atenciones Médicas Registradas	46
Figura 2.8. Diagrama de Casos de Uso: Atenciones Médicas Por Entregar Medicamentos	47
Figura 2.9. Diagrama de Casos de Uso: Atenciones Médicas Finalizadas	47
Figura 2.10. Diagrama de Actividades: Nuevo Stock de Medicamento	48
Figura 2.11. Diagrama de Actividades: Lista de Medicamentos en Stock.....	49
Figura 2.12. Diagrama de Actividades: Historial Farmacéutico de Medicamento	50
Figura 2.13. Diagrama de Actividades: Nueva Atención Médica	52
Figura 2.14. Diagrama de Actividades: Atenciones Médicas Registradas	53
Figura 2.15. Diagrama de Actividades: Atenciones Médicas Por Entregar Medicamentos	55
Figura 2.16. Diagrama de Actividades: Atenciones Médicas Finalizadas	56
Figura 2.17. Diagrama de Clases	57
Figura 2.18. Sketch Registro de Usuario	58
Figura 2.19. Sketch Usuarios Registrados	59
Figura 2.20. Sketch Inicio de Sesión de Usuario.....	59
Figura 2.21. Sketch Menú Administrador.....	60
Figura 2.22. Sketch Nuevo Stock de Medicamento	61
Figura 2.23. Sketch Lista de Medicamentos en Stock	62
Figura 2.24. Sketch Historial Farmacéutico de Medicamento.....	63
Figura 2.25. Sketch Menú Médico	63
Figura 2.26. Sketch Nueva Atención Médica.....	64
Figura 2.27. Sketch Atenciones Médicas Registradas	65
Figura 2.28. Sketch Menú Dependiente de Farmacia	66

Figura 2.29. Sketch Atenciones Médicas Por Entregar Medicamentos	67
Figura 2.30. Sketch Atenciones Médicas Finalizadas	68
Figura 2.31. Sketch Datos Personales de Usuario	69
Figura 2.32. Instancia de EC2 Seleccionada.....	71
Figura 2.33. Volumen de Instancia EC2	71
Figura 2.34. Instancia de EC2 Creada.....	71
Figura 2.35. Instancia de RDS Seleccionada	72
Figura 2.39. Instancia de RDS Creada	73
Figura 2.40. Conexión Local de Base de Datos	73
Figura 2.41. Archivo package.json.....	75
Figura 2.42. Servidor Local Ejecutado en el Puerto 8080.....	81
Figura 2.43. Obtención de Especialidades Médicas Registradas en Base de Datos Local a través de un Navegador Web.....	82
Figura 2.44. Credenciales de Acceso a Servicio de Base de Datos RDS	82
Figura 2.45. Escritorio Remoto Asignado por AWS	83
Figura 2.46. Obtención de Especialidades Médicas Registradas en Base de Datos de AWS a través de un Navegador Web	84
Figura 2.47. Plantillas para la creación de una Activity	84
Figura 2.48. Configuración de Proyecto en Android Studio	85
Figura 2.49. Versiones de Android.....	85
Figura 2.50. Interfaz Gráfica de Activity Inicio de Sesión	88
Figura 3.1. Registro de Usuario Administrador.....	92
Figura 3.2. Nuevo Usuario Administrador Registrado en la Base De Datos	92
Figura 3.3. Registro de Usuario Médico	93
Figura 3.4. Nuevo Usuario Médico Registrado en la Base De Datos	93
Figura 3.5. Usuario Médico Registrado Pendiente de Validación.....	94
Figura 3.6. Cuadro de Dialogo de Activación de Usuario Médico.....	94
Figura 3.7. Usuario Médico Activado	95
Figura 3.8. Actualización de Estado de Médico a “Activo” en la tabla tbl_Medico en la Base de Datos	95
Figura 3.9. Registro de Usuario Dependiente de Farmacia	96
Figura 3.10. Nuevo Usuario Dependiente de Farmacia Registrado en la Base De Datos 96	
Figura 3.11. Usuario Dependiente de Farmacia Registrado Pendiente de Validación	97
Figura 3.12. Cuadro de Dialogo de Activación de Usuario Dependiente de Farmacia	97
Figura 3.13. Usuario Dependiente de Farmacia Activado.....	98
Figura 3.14. Actualización de Estado de Dependiente de Farmacia a “Activo” en la tabla tbl_Dependiente_Farmacia en la Base de Datos	98
Figura 3.15. Inicio de Sesión de Usuario Administrador	99

Figura 3.16. Mensaje de Inicio de Sesión de Usuario Administrador Correcto	99
Figura 3.17. Inicio de Sesión de Usuario Médico	100
Figura 3.18. Mensaje de Inicio de Sesión de Usuario Médico Correcto	100
Figura 3.19. Inicio de Sesión de Usuario Dependiente de Farmacia.....	101
Figura 3.20. Mensaje de Inicio de Sesión de Usuario Dependiente de Farmacia Correcto	101
Figura 3.21. Segundo Usuario Médico Registrado en la Base De Datos	102
Figura 3.22. Médico Visualiza sus Datos Personales	102
Figura 3.23. Médico Actualiza sus Datos Personales	103
Figura 3.24. Actualización de Apellido en la tabla tbl_Medico en la Base de Datos ..	103
Figura 3.25. Médico Actualiza su Contraseña	104
Figura 3.26. Actualización de Contraseña en la tabla tbl_Medico en la Base de Datos 104	
Figura 3.27. Médico Desactiva su Cuenta	105
Figura 3.28. Actualización de Estado de Usuario a Desactivo en la tabla tbl_Medico en la Base de Datos	105
Figura 3.29. Médico Elimina su Cuenta	106
Figura 3.30. Eliminación de Médico en la tabla tbl_Medico en la Base de Datos	106
Figura 3.31. Ingreso de Nuevo Tipo de Medicamento	107
Figura 3.32. Ingreso de Nuevo Medicamento y su Concentración	108
Figura 3.33. Ingreso de Unidades de Nuevo Medicamento	108
Figura 3.34. Creación de Stock de Medicamento.....	109
Figura 3.35. Nuevo Tipo de Medicamento Registrado en la Base De Datos	109
Figura 3.36. Nuevo Medicamento Registrado en la Base De Datos.....	109
Figura 3.37. Nuevo Stock de Medicamento Registrado en la Base De Datos	110
Figura 3.38. Cuatro Nuevos Tipos de Medicamentos Registrados en la Base De Datos 110	
Figura 3.39. Nueve Nuevos Medicamentos Registrados en la Base De Datos	110
Figura 3.40. Nueve Nuevos Stocks de Medicamentos Registrados en la Base De Datos.....	110
Figura 3.41. Lista de Medicamentos en Stock Registrados	111
Figura 3.42. Confirmación de Eliminación de Stock de Medicamento.....	112
Figura 3.43. Eliminación de Stock de Medicamento en la tabla tbl_Medicamento_Stock en la Base de Datos	112
Figura 3.44. Lista de Medicamentos en Stock Actualizada.....	113
Figura 3.45. Ingreso de Cédula de Nuevo Paciente.....	114
Figura 3.46. Ingreso de Datos Personales de Nuevo Paciente	114
Figura 3.47. Medicamento Agregado a la Receta	115
Figura 3.48. Cuatro Medicamentos Más Agregados a la Receta	115

Figura 3.49. Medicamento No Agregado a la Receta por Falta de Stock.....	116
Figura 3.50. Paciente Nuevo Registrado en la Base De Datos	116
Figura 3.51. Atención Médica Nueva Registrada en la Base De Datos	116
Figura 3.52. Medicamentos de Receta Registrados en la Base de Datos	117
Figura 3.53. Segundo Paciente Registrado en la Base De Datos	117
Figura 3.54. Segunda Atención Médica Registrada en la Base De Datos	117
Figura 3.55. Medicamentos de Segunda Receta Registrados en la Base de Datos..	117
Figura 3.56. Buscar Paciente Registrado por Cédula	118
Figura 3.57. Paciente Registrado y Atención Médica Pendiente	118
Figura 3.58. Detalles de Atención Médica Pendiente de Entrega de Medicamentos.	119
Figura 3.59. Cuadro de Dialogo de Confirmación de Entrega de Medicamento	120
Figura 3.60. Registro de Fecha de Entrega de Medicamento de Receta	120
Figura 3.61. Fecha de Entrega Registrada en Todos los Medicamentos de Receta .	121
Figura 3.62. Mensaje de Entrega Completa de Medicamentos de Receta	122
Figura 3.63. Actualización del Estado de la Primera Atención Médica a “Entregado Completo” en la Base de Datos	122
Figura 3.64. Actualización de Entrega Total de Medicamentos de la Primera Atención Médica en la Base de Datos	122
Figura 3.65. Medicamentos Parcialmente Entregados de Segunda Atención Médica	123
Figura 3.66. Medicamentos No Entregados Por Falta de Stock	124
Figura 3.67. Actualización del Estado de la Segunda Atención Médica a “Parcialmente Entregado” en la Base de Datos	124
Figura 3.68. Actualización de Entrega Parcial de Medicamentos de la Segunda Atención Médica en la Base de Datos	125
Figura 3.69. Lista de Atenciones Médicas Registradas	125
Figura 3.70. Detalles de Atención Médica con Estado “Parcialmente Entregado”	126
Figura 3.71. Cuadro de Dialogo de Actualización de Medicamento de Receta	127
Figura 3.72. Ingreso de Datos de Medicamento Equivalente	127
Figura 3.73. Medicamento de Receta de Actualizado por Equivalente	128
Figura 3.74. Actualización de Medicamento de Receta por Equivalente de Segunda Atención Médica en la Base de Datos	128
Figura 3.75. Cuadro de Dialogo de Eliminación de Medicamento de Receta	129
Figura 3.76. Medicamento de Receta Eliminado	129
Figura 3.77. Eliminación de Medicamento de Receta de Segunda Receta en la Base de Datos.....	130
Figura 3.78. Registro de Entrega de Medicamento de Receta Actualizado	130
Figura 3.79. Actualización del Estado de la Segunda Atención Médica a “Entregado Completo” en la Base de Datos	131
Figura 3.80. Actualización de Entrega Total de Medicamentos de la Segunda Atención Médica en la Base de Datos	131

Figura 3.81. Lista de Atenciones Médicas Finalizadas	131
Figura 3.82. Detalles de Atención Médica Finalizada	132
Figura 3.83. Datos de Medicamento Para Historial Farmacéutico.....	132
Figura 3.84. Selección de Tipo de Historial: Total	133
Figura 3.85. Historial Farmacéutico de Medicamento	133
Figura 3.86. Mensaje de Creación de PDF	134
Figura 3.87. Archivo PDF Creado en el Almacenamiento del Dispositivo	135
Figura 3.88. Información Contendida en el Archivo PDF	135

ÍNDICE DE TABLAS

Tabla 1.1. Métodos de HTTP	9
Tabla 2.1. Tablero Kanban Inicial	25
Tabla 2.2. Resultados de Entrevista a Administrador (Parte 1 de 3)	26
Tabla 2.3. Resultados de Entrevista a Administrador (Parte 2 de 3)	27
Tabla 2.4. Resultados de Entrevista a Administrador (Parte 3 de 3)	28
Tabla 2.5. Resultados de Entrevistas a Médicos (Parte 1 de 3)	28
Tabla 2.6. Resultados de Entrevistas a Médicos (Parte 2 de 3)	29
Tabla 2.7. Resultados de Entrevistas a Médicos (Parte 3 de 3)	30
Tabla 2.8. Resultados de Entrevistas a Dependientes de Farmacia (Parte 1 de 3).....	30
Tabla 2.9. Resultados de Entrevistas a Dependientes de Farmacia (Parte 2 de 3).....	31
Tabla 2.10. Resultados de Entrevistas a Dependientes de Farmacia (Parte 3 de 3)...	32
Tabla 2.11. Requerimientos Funcionales	33
Tabla 2.12. Requerimientos No Funcionales.....	34
Tabla 2.13. Estimación RF Módulo Registro de Usuario.....	37
Tabla 2.14. Estimación RF Módulo Inicio de Sesión de Usuario	37
Tabla 2.15. Estimación RF Módulo Datos Personales de Usuario	37
Tabla 2.16. Estimación RF Módulo Nuevo Stock de Medicamento	38
Tabla 2.17. Estimación RF Módulo Lista de Medicamentos en Stock	38
Tabla 2.18. Estimación RF Módulo Historial Farmacéutico de Medicamento	38
Tabla 2.19. Estimación RF Módulo Nueva Atención Médica	39
Tabla 2.20. Estimación RF Módulo Atenciones Médicas Registradas.....	39
Tabla 2.21. Estimación RF Módulo Atenciones Médicas Por Entregar Medicamentos	39
Tabla 2.22. Estimación RF Módulo Atenciones Médicas Finalizadas.....	40
Tabla 2.23. Casos de Uso de Cada Módulo	44
Tabla 2.24. Primera Actualización de Tablero Kanban	70
Figura 2.36. Modelo Free Tier de Instancia RDS	72
Figura 2.37. Identificación de Instancia RDS.....	72
Figura 2.38. Volumen de Instancia RDS.....	73
Tabla 3.1. Segunda Actualización de Tablero Kanban	91
Tabla 3.2. Resultado Encuesta de Validación: Administrador (Parte 1 de 2).....	136
Tabla 3.3. Resultado Encuesta de Validación: Administrador (Parte 2 de 2).....	137
Tabla 3.4. Resultado Encuesta de Validación: Médico (Parte 1 de 2).....	137
Tabla 3.5. Resultado Encuesta de Validación: Médico (Parte 2 de 2).....	138
Tabla 3.6. Resultado Encuesta de Validación: Dependiente de Farmacia (Parte 1 de 2)	139

Tabla 3.7. Resultado Encuesta de Validación: Dependiente de Farmacia (Parte 2 de 2)
140

Tabla 3.8. Tablero Kanban Final..... 141

ÍNDICE DE CÓDIGOS

Código 1.1. Ejemplo de Interfaz con Método con Operación HTTP	19
Código 1.2. Ejemplo de Cliente Retrofit.....	19
Código 1.3. Ejemplo de llamada Call	19
Código 1.4. Ejemplo de Anotación GET de HTTP.....	19
Código 1.5. Ejemplo de Método con Anotación POST.....	20
Código 2.1. Creación de Base de Datos.....	74
Código 2.2. Creación de Tabla tbl_Tipo_Medicamento	74
Código 2.3. Creación de Tabla tbl_Medicamento.....	74
Código 2.4. Configuración del package.json	75
Código 2.5. Configuración del Servidor	76
Código 2.6. Ruta de Servicio obtenerEspecialidadesMedicas.....	77
Código 2.7. Conexión Base de Datos y consulta SQL obtenerEspecialidadesMedicas 78	
Código 2.8. Clase Medico	79
Código 2.9. Librerías añadidas a Gradle (Parte 1 de 2).....	80
Código 2.10. Creación de Cliente Retrofit	80
Código 2.11. Declaración de Métodos con las Operaciones HTTP	81
Código 2.12. Establecimiento de Conexión con la Base de Datos RDS de AWS	83
Código 2.13. Librerías añadidas a Gradle (Parte 2 de 2).....	86
Código 2.14. Permisos de Usuario en el AndroidManifest.....	86
Código 2.15. Activity en el AndroidManifest	86
Código 2.16. Implementación de Parte Gráfica de Activity Inicio de Sesión (Parte 1 de 2) 87	
Código 2.17. Implementación de Parte Gráfica de Activity Inicio de Sesión (Parte 2 de 2) 88	
Código 2.18. Validación de Cédula de Identidad.....	89
Código 2.19. Método de Registro de Usuario Médico.....	90

RESUMEN

El objetivo de este proyecto de titulación es el desarrollo de un prototipo de aplicación móvil para gestionar registros farmacéuticos de medicamentos del Centro Integral de Especialidades Centro, el cual dispone de tres usuarios: Administrador, Médico y Dependiente de Farmacia. El administrador se encarga de registrar los datos de un nuevo stock de medicamentos adquiridos, el médico luego de cada atención médica registra uno o varios medicamentos requeridos por cada paciente atendido, así como la cantidad necesaria para su tratamiento, el dependiente de farmacia encargado de despachar los medicamentos a los pacientes, visualiza la información registrada por cada médico en las atenciones médicas realizadas y registra la entrega de medicamentos a cada paciente. Con estos registros de medicamentos entrantes y salientes, el administrador visualiza los datos correspondientes al historial farmacéutico de cada medicamento, facilitando la verificación del stock de medicamentos existentes en el centro de salud.

La aplicación móvil permite el registro de cada usuario que va a hacer uso de la misma. Adicionalmente, incluye el desarrollo de un servicio web y una base de datos para almacenar y gestionar la información, ambos alojados en un proveedor de cloud computing.

El presente documento está compuesto por cuatro capítulos en los cuales se describe el procedimiento seguido para el desarrollo del prototipo:

En el Capítulo 1, se describen los fundamentos teóricos sobre temas necesarios para el proyecto: base de datos, servicio web, cloud computing, smartphone, aplicaciones móviles, Android y la metodología de desarrollo Kanban.

En el Capítulo 2, se detalla la metodología utilizada y consta de dos partes: diseño e implementación.

En el Capítulo 3, se presentan las pruebas de funcionamiento del prototipo.

En el Capítulo 4, se presentan las conclusiones y recomendaciones obtenidas en el desarrollo del proyecto.

PALABRAS CLAVE: aplicación móvil, base de datos, servicio web, cloud computing, Android Studio.

ABSTRACT

The objective of this degree project is the development of a prototype mobile application to manage pharmaceutical drug records of the Centro Integral de Especialidades Centro, which has three users: Administrator, Doctor and Pharmacy Clerk, the administrator is in charge of registering the data of a new stock of medicines acquired, the doctor after each medical care, registering one or more medicines required by each patient attended, as well as the amount necessary for their treatment, the pharmacy clerk, who is in charge of dispensing the medicines to the patients, it visualizes the information registered by each doctor of the medical care operated and it is in charge of registering the delivery of medicines to each patient. With these records of incoming and outgoing drugs, the administrator visualizes the data corresponding to the pharmaceutical history of each drug, facilitating the verification of the stock of drugs affected in the health center.

The mobile application allows the registration of each user who is going to use it. Additionally, it includes the development of a web service and a database to store and manage information, both hosted by a cloud computing provider.

The present document is made up of four chapters in which the procedure carried out for the development of the prototype is described:

In Chapter 1, the theoretical foundations on topics necessary for the project are described: database, web service, cloud computing, smartphone, mobile applications, Android and the Kanban development methodology.

In Chapter 2, details the methodology used and consists of two parts: design and implementation.

In Chapter 3, the functional tests of the prototype are presented.

In Chapter 4, the conclusions and recommendations obtained in the development of the project are described.

KEY WORDS: mobile application, database, web service, cloud computing, Android Studio

1 INTRODUCCIÓN

El Centro Integral de Especialidades Centro gestiona registros detallados con información correspondiente a la cantidad y la fecha en la que los medicamentos ingresan y salen de sus instalaciones. Estos registros son de mucha utilidad para conocer si un medicamento existe o no en stock, y así poder adquirir un nuevo stock de medicamentos en caso de faltantes. Esta información debe tener un almacenamiento apropiado que facilite el acceso por parte de las autoridades de dichos centros de salud.

El centro de salud registra dicha información en documentos manuscritos, lo cual puede presentar inconvenientes en el contenido como: escritura ilegible, datos faltantes, registros incompletos, generando dificultades para acceder a la información.

En la actualidad el smartphone se ha convertido en objeto imprescindible en nuestro día a día, debido a su capacidad de procesamiento y fácil acceso a diversos recursos a través de Internet, lo que permite a los usuarios desde cualquier ubicación acceder a la información que requieren.

Los servicios en la nube no requieren adquirir e instalar ningún tipo de hardware y generan soluciones a los requerimientos de las aplicaciones móviles, permitiendo un desarrollo ágil de las mismas. Además, se puede acceder a dichos servicios desde cualquier lugar y tener acceso a toda la información.

En Ecuador se han desarrollado aplicaciones móviles para los centros de salud enfocados en agendar citas médicas y realizar pagos en línea, por lo que el presente proyecto se presenta como una solución tecnológica adaptable a los requerimientos del Centro Integral de Especialidades Centro y su finalidad es desarrollar un prototipo de aplicación móvil para gestionar registros farmacéuticos de medicamentos, que permita generar una interacción entre tres usuarios: administrador, personal médico y personal de farmacia.

El proyecto está conformado por una aplicación móvil para Android, un servicio web Rest y una base de datos. El servicio web y la base de datos se encuentran alojados en la nube.

1.1 OBJETIVOS

El objetivo general de este proyecto de titulación es desarrollar un prototipo de una aplicación para teléfonos inteligentes que permita gestionar el registro farmacéutico de medicamentos.

Los objetivos específicos de este proyecto de titulación son:

- Analizar los fundamentos teóricos necesarios para el desarrollo del proyecto.
- Diseñar el prototipo de la aplicación móvil.
- Implementar el prototipo de la aplicación móvil.
- Analizar los resultados obtenidos en las pruebas de funcionamiento del prototipo.

1.2 ALCANCE

El presente proyecto presenta el diseño e implementación de un prototipo de aplicación móvil para teléfonos inteligentes con Sistema Operativo Android para el “Centro Integral de Especialidades Centro (CIEC)”. El aplicativo dispone de tres tipos de usuarios: USUARIO ADMINISTRADOR, USUARIO MÉDICO Y USUARIO DEPENDIENTE DE FARMACIA. Además, el prototipo tiene una base de datos y un servicio web, los cuales son utilizados por los tres usuarios.

Para el alojamiento de la base de datos y el servicio web se utiliza Cloud Computing a través del proveedor Amazon Web Services (AWS).

En forma general el prototipo considera los siguientes requerimientos mínimos:

La primera vez que se utiliza la aplicación móvil, los tres tipos de usuario: ADMINISTRADOR, MÉDICO Y DEPENDIENTE DE FARMACIA, se registran en la misma, ingresando sus datos personales, así como una contraseña.

Para iniciar sesión e ingresar a las funcionalidades de la aplicación móvil, los tres tipos de usuario deben ingresar sus credenciales: cédula y contraseña.

La aplicación móvil utiliza el servicio web para el acceso y almacenamiento de información en la base de datos, logrando que los tres tipos de usuario: ADMINISTRADOR, MÉDICO Y DEPENDIENTE DE FARMACIA, estén relacionados entre sí.

Cada usuario realiza las siguientes actividades:

- **MÉDICO:** realiza los procesos de creación, almacenamiento y eliminación de la información correspondiente a cada una de las atenciones médicas llevadas a cabo, especificando los datos del paciente y la receta en la que se incluye uno o varios medicamentos requeridos por dicho paciente, así como la cantidad requerida de cada medicamento. Dicha información se almacena en la base de datos alojada en la nube. Esta información es visualizada por el usuario Dependiente de Farmacia. En caso de no existir el stock de un medicamento de una receta específica, el médico visualiza esta información y procede a actualizar la receta reemplazando dicho medicamento por otro equivalente o eliminando dicho medicamento de la receta en caso de ser necesario y generando una segunda receta manuscrita para que el paciente pueda adquirir el medicamento en una farmacia externa.
- **DEPENDIENTE DE FARMACIA:** visualiza la información de la o las atenciones médicas recibidas por un paciente específico; al seleccionar una de ellas, se detalla la información de los medicamentos recetados por el médico que le atendió. En base a la cantidad solicitada se verifica la existencia o no de los medicamentos en stock. En caso de existir la cantidad requerida, se registra dicha entrega en la aplicación móvil. En caso de no existir la cantidad requerida, la atención médica se actualiza para que el médico tratante pueda reemplazar el medicamento no existente por otro equivalente o eliminarlo de la receta. Otra de las funciones del dependiente de farmacia es visualizar el listado de atenciones médicas en las que se ha finalizado con el registro de la entrega de todos los medicamentos.
- **ADMINISTRADOR:** usuario que se encarga de realizar los procesos de creación, almacenamiento y eliminación de los datos correspondientes a cada stock nuevo referente a un medicamento específico que ingresa al CIEC, especificando el tipo de medicamento, el nombre del medicamento, la concentración y la cantidad total de ingreso. Dicha información se almacena en la base de datos alojada en la nube. Además, el Administrador gestiona el registro correspondiente al historial farmacéutico entrante y saliente del CIEC, para lo cual selecciona un medicamento específico y elige una de las tres modalidades, que puede ser total, por año o por año y mes. Permitiendo la verificación del stock del medicamento existente en el centro de salud.

En la Figura 1.1, se muestra el esquema general de funcionamiento del prototipo de aplicación móvil para teléfonos inteligentes con Android.

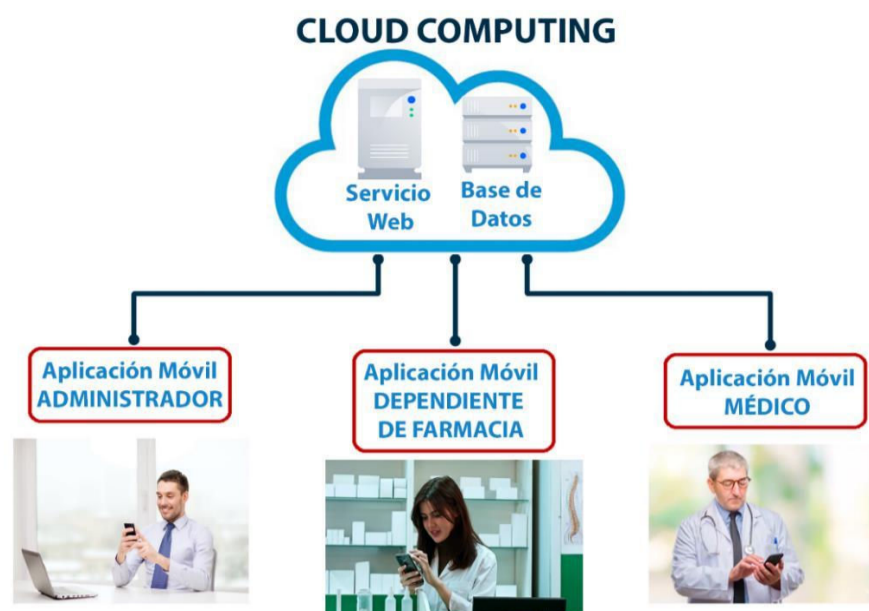


Figura 1.1. Esquema de funcionamiento del Proyecto

La aplicación móvil dispone de las siguientes características:

- **Registro:** los usuarios se registran con sus datos personales, se debe incluir una contraseña para poder tener acceso a la aplicación móvil.
- **Inicio de Sesión:** mediante el ingreso de la cédula y una contraseña previamente registrada por el usuario, se accede a las funciones de la aplicación de acuerdo al tipo de usuario correspondiente.
- **Datos de Paciente:** el médico registra los datos personales del paciente que ha recibido una atención médica, en caso de haber sido registrado por una atención médica previa, se buscan los datos del paciente en la base de datos alojada en la nube.
- **Receta:** el médico luego de registrar los datos personales del paciente, selecciona uno o varios medicamentos requeridos por el paciente y registra en la aplicación móvil dicha información.
- **Medicamentos Por Entregar:** el dependiente de farmacia al ingresar la cédula de un paciente, se muestra la lista de atenciones médicas recibidas por dicho paciente y al seleccionar una de ellas se despliega la información correspondiente a uno o varios medicamentos requeridos por dicho paciente. Si existen los medicamentos en stock, se registra la entrega en la aplicación móvil, caso contrario la atención medica se actualiza para que el médico tratante pueda reemplazar el medicamento no existente por otro equivalente o eliminarlo de la receta.
- **Medicamentos Entregados:** el dependiente de farmacia visualiza la lista de atenciones médicas que ya han sido finalizadas. Al seleccionar una de ellas se

despliega los detalles de dicha atención médica con la lista de medicamentos que ya fueron entregados al paciente.

- **Atenciones Médicas Realizadas:** el médico luego de registrar una atención médica tiene la capacidad de visualizar dicha atención médica con todos sus detalles para saber los medicamentos que ya registran entrega y los que no. En caso de haber uno o varios medicamentos no entregados puede cambiar cada medicamento por otro equivalente o eliminar los medicamentos de dicha receta.
- **Ingreso de Medicamentos:** el administrador registra el ingreso de un nuevo stock de medicamentos, especificando el tipo de medicamento, nombre de medicamento, cantidad y concentración.
- **Historial Farmacéutico de Medicamentos:** el administrador selecciona un medicamento específico y elige una de las modalidades: total, por año o por año y mes, y se genera en formato PDF el registro del historial de ingreso y salida del medicamento seleccionado.

1.3 MARCO TEÓRICO

1.3.1 BASE DE DATOS

La base de datos es una recopilación sistemática de datos, los cuales pueden ser gestionados [1].

1.3.1.1 SISTEMA DE GESTIÓN DE BASES DE DATOS (DBMS)

Un sistema de gestión de bases de datos o DBMS es un software que proporciona la interfaz entre los usuarios, aplicaciones y una o más bases de datos [1]. El objetivo principal de un DBMS es proporcionar a los usuarios una forma eficiente de almacenar y recuperar información de la base de datos. Los sistemas de bases de datos están diseñados para administrar grandes cantidades de información [2].

La gestión de datos implica definir estructuras para el almacenamiento de información y proporcionar mecanismos para la manipulación de la información. Además, el sistema debe garantizar la seguridad de la información almacenada, a pesar de bloqueos del sistema o intentos de acceso no autorizado. Si los datos se van a compartir entre varios usuarios, el sistema debe evitar posibles resultados anómalos [3].

Ejemplos de DBMS [4]:

- MySQL.
- SQL Server.
- Oracle.
- PostgreSQL.
- SQLite
- MariaDB, etc.

1.3.1.2 MODELO RELACIONAL

El modelo relacional emplea un conjunto de tablas, cada tabla tiene varias columnas y cada columna tiene un nombre único. En este modelo las tablas comparten alguna columna entre ellas, estas columnas compartidas van a servir para establecer relaciones entre las tablas [3].

El modelo relacional está basado en registros. Cada tabla (Figura 1.2) contiene registros de un tipo particular, cada tipo de registro define un número fijo de campos o atributos y las columnas de la tabla corresponden a los atributos del tipo de registro [3].

<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>salary</i>
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Figura 1.2. Tabla de Base de Datos

Las relaciones que se establecen entre dos tablas en una base de datos relacional pueden ser de tres tipos [2]:

- Uno a uno: se establecen entre una entidad de una tabla y otra entidad de otra tabla.
- Uno a varios: se establecen entre varias entidades de una tabla y una entidad de otra tabla.
- Varios a varios: se establecen entre varias entidades de cada una de las tablas.

Llave Primaria:

Una llave primaria es un campo en una tabla que identifica de manera única cada registro de dicha tabla, por lo que no se puede tener dos o más registros que tengan el mismo valor de ese campo. La llave primaria no puede tener valores NULL [5].

Llave Foránea:

También conocida como llave de referencia, es un campo en una tabla que hace referencia a la llave primaria en otra tabla, es decir, es una llave utilizada para vincular dos tablas [6].

La tabla que contiene la llave foránea se llama tabla secundaria y la tabla que contiene la llave primaria se denomina tabla principal o referenciada [6].

1.3.1.3 LENGUAJE DE CONSULTA ESTRUCTURADO (SQL)

SQL es el lenguaje estándar para los sistemas de gestión de bases de datos relacionales. Las sentencias SQL se utilizan para realizar tareas como crear, insertar, actualizar, recuperar o eliminar datos de una base de datos. Entre las sentencias más importantes se encuentran [7]:

- CREATE
- INSERT
- UPDATE
- SELECT
- DELETE

Algunos sistemas comunes de administración de bases de datos relacionales que utilizan SQL son: Oracle, Microsoft SQL Server, Access, MySQL, etc. [2].

1.3.2 SERVICIO WEB

Un servicio web es una aplicación autónoma que se puede describir, publicar, localizar e invocar a través de una red [8], es decir, es un tipo de aplicación web entregada a través de HTTP (Hyper Text Transport Protocol), permitiendo aumentar la flexibilidad de los procesos empresariales al integrarse con aplicaciones que de otra forma no se comunicarían, por lo tanto, un servicio web es un servicio distribuido, cuyos

componentes se pueden implementar y ejecutar en distintos dispositivos. Los servicios web se pueden dividir aproximadamente en dos grupos: SOAP y REST [9]. Para este proyecto se implementará los servicios web de tipo REST.

1.3.2.1 REST (REPRESENTATIONAL STATE TRANSFER)

REST es una interfaz entre sistemas que utilizan HTTP para obtener datos o generar operaciones sobre esos datos en los formatos posibles como XML y JSON [10]. REST se utiliza para crear servicios web ligeros, mantenibles y escalables [11]. Las principales características de REST son:

- **Todo es un recurso:**

Los datos se representan por un formato específico y no por un archivo físico. Cada pieza de datos disponible en Internet tiene un formato que podría describirse por un tipo de contenido [12].

- **Cada recurso es identificable por un identificador único:**

Los recursos en Internet son accesibles a través de una URI permitiendo ser identificados de manera única, el URI mantiene los datos autodescriptivos y facilita un mayor desarrollo en ellos. Además, el uso de URI legibles ayuda a reducir el riesgo de errores lógicos [12].

Ejemplos de URIs:

- <http://www.misdatos.com/imagenes/playa/2016/verano>
- <http://www.misdatos.com/videos/sierra/2017/invierno>
- <http://www.misdatos.com/datos/documentos/balance/2018/agosto>
- <http://www.misdatos.com/datos/archivos/2019>

- **Utiliza los métodos HTTP estándar:**

Los principales métodos de HTTP (Tabla 1.1) utilizados en REST tienen similitudes características con las bases de datos SQL donde el idioma nativo para la manipulación de datos es CRUD (abreviatura para Crear, Leer, Actualizar y Eliminar). De la misma manera, si se aplica los principios REST correctamente se deben usar los métodos HTTP [12].

Tabla 1.1. Métodos de HTTP

HTTP	Acción	Código de respuesta
GET	Solicitar un recurso existente	"200: OK" si el recurso existe, "404: Not Found" si no existe el recurso, y "500: Internal Server Error" para otros errores.
POST	Crear un nuevo recurso	"201 CREATED" si se crea un nuevo recurso y "500: Internal Server Error" para otros errores.
PUT	Actualizar un recurso existente	"200: OK" si el recurso ha sido actualizado con éxito, "404: Not Found" si el recurso a ser actualizado no existe y "500: Internal Server Error" para otros errores.
DELETE	Eliminar un recurso existente	"200: OK" si el recurso ha sido eliminado con éxito, "404: Not Found" si el recurso a ser eliminado no existe y "500: Internal Server Error" para otros errores.

- **Los recursos pueden tener múltiples representaciones:**

Una característica clave de un recurso es que pueden estar representados en una forma diferente al que está almacenado, por lo tanto, se puede solicitar o publicar en diferentes representaciones [12].

1.3.3 CLOUD COMPUTING

La computación en la nube (Cloud Computing) es la entrega a pedido de recursos de Tecnologías de Información (TI) a través de Internet con precios de pago por uso, en lugar de comprar, poseer y mantener servidores y centros de datos físicos, se puede acceder a servicios tecnológicos, como potencia informática, almacenamiento y bases de datos, según sea necesario y de un proveedor de la nube [13].

Las organizaciones de todo tipo, tamaño e industria están utilizando la nube para una amplia variedad de casos de uso, como copia de seguridad de datos, recuperación ante desastres, correo electrónico, escritorios virtuales, desarrollo y prueba de software y aplicaciones web orientadas al cliente. Por ejemplo, las compañías de atención médica están utilizando la nube para desarrollar tratamientos más personalizados para los pacientes, las compañías de servicios financieros están utilizando la nube para impulsar la detección y prevención de fraudes en tiempo real y los fabricantes de videojuegos están utilizando la nube para ofrecer juegos en línea a millones de jugadores de todo el mundo [14].

1.3.3.1 TIPOS DE CLOUD COMPUTING

- **Público:**

Los clouds públicos pertenecen y son administrados por empresas que ofrecen a través de una red pública acceso rápido a recursos informáticos asequibles. Con los servicios de cloud público los usuarios no necesitan adquirir hardware, ni software ni infraestructura de soporte, ya que pertenece a los proveedores y lo gestionan ellos [14].

- **Privado:**

Un cloud privado se trata de una infraestructura que utiliza únicamente una única organización, ya sea gestionada internamente o por terceros y alojada internamente o externamente. Los clouds privados aprovechan la eficiencia del cloud, a la vez que ofrecen un mayor control de los recursos y evitan la multitenencia [14].

- **Híbrido:**

Un cloud híbrido utiliza una base de cloud privado combinada con la integración estratégica y el uso de servicios cloud público. La mayoría de las empresas con clouds privados gestionan cargas de trabajo en todos los centros de datos, clouds privados y clouds públicos, creando así clouds híbridos [14].

1.3.3.2 TIPOS DE SERVICIOS

Los tres principales tipos de computación en la nube son [13]:

- Infraestructura como servicio.
- Plataforma como servicio.
- Software como servicio.

Cada tipo de computación en la nube proporciona diferentes niveles de control, flexibilidad y administración, de adecuado a las necesidades del cliente [13].

- **Infraestructura como servicio (IaaS):**

IaaS proporciona acceso a funciones de red, computadoras (virtuales o en hardware dedicado) y espacio de almacenamiento de datos. Además, ofrece flexibilidad y control de gestión sobre los recursos de TI [13].

- **Plataforma como servicio (PaaS):**

PaaS elimina la necesidad de administrar la infraestructura (generalmente hardware y sistemas operativos), permitiendo concentrarse en la implementación y administración de diversas aplicaciones siendo más eficiente, ya que no necesita preocuparse por la adquisición de recursos, la planificación de la capacidad, el mantenimiento del software, la aplicación de parches o cualquiera de los otros trabajos pesados no diferenciados involucrados en la ejecución de las aplicaciones [13].

- **Software como servicio (SaaS):**

SaaS proporciona un producto completo que es ejecutado y administrado por el proveedor de servicios, en la mayoría de los casos se refiere a aplicaciones de usuario final (como el correo electrónico). Con SaaS no hay que centrarse en cómo se mantiene el servicio o cómo se gestiona la infraestructura subyacente, solo en cómo usar un software en particular [13].

1.3.3.3 BENEFICIOS DE CLOUD COMPUTING

- **Agilidad:** la nube brinda fácil acceso a una amplia gama de tecnologías permitiendo innovar más rápido y construir casi cualquier cosa que se pueda imaginar, puede aumentar rápidamente los recursos según sea necesario, desde servicios de infraestructura, como informática, almacenamiento y bases de datos, hasta Internet de las cosas, aprendizaje automático, etc. [13].

Puede implementar servicios de tecnología en cuestión de minutos y pasar de la idea a la implementación más rápido que antes, esto da la libertad de experimentar, probar nuevas ideas para diferenciar las experiencias de los clientes y transformar los negocios [13].

- **Elasticidad:** con la computación en la nube se puede aprovisionar la cantidad de recursos realmente necesarios, ya que se puede aumentar o reducir estos recursos al instante para aumentar y reducir la capacidad a medida que cambian las necesidades de un negocio [13].

- **Ahorro de costos:** la nube permite generar gastos sólo por recursos a medida que se consumen [13].

- **Implementación global en minutos:** la infraestructura permite implementar y expandir una aplicación en múltiples ubicaciones físicas en todo el mundo con solo unos pocos clics. Poner las aplicaciones más cerca de los usuarios finales reduce la latencia y mejora su experiencia [13].

1.3.3.4 AMAZON WEB SERVICES

Amazon Web Services (AWS) es una división de Amazon y es una plataforma en la nube que ofrece múltiples servicios completos de centros de datos a nivel mundial, proporciona servidores, almacenamiento, redes, informática remota, correo electrónico, desarrollo móvil y seguridad, aloja sitios web populares como Netflix e Instagram. Entre los múltiples servicios ofrecidos por AWS se encuentran los principales que son: Computación y Base de datos [15].

Servicio de Computación:

A nivel de computación uno de sus principales servicios es Amazon Elastic Compute Cloud (Amazon EC2).

- **Amazon Elastic Compute Cloud (Amazon EC2):** es un servicio que proporciona capacidad informática escalable en la nube de AWS, elimina la necesidad de invertir en hardware físico, permitiendo desarrollar e implementar aplicaciones más rápido. Se puede iniciar muchos o pocos servidores virtuales como sea necesario, configurar la seguridad y las redes, y administrar el almacenamiento. Permite realizar un escalamiento hacia arriba o hacia abajo para manejar cambios en los requisitos o picos de tráfico y sus características son [16]:
 - Entornos virtuales de computación conocidos como instancias.
 - Diversas configuraciones de CPU, memoria, almacenamiento y capacidad de red para las instancias.
 - Volúmenes de almacenamiento de datos temporales que se eliminan cuando detiene o finaliza una instancia.
 - Múltiples ubicaciones físicas para los recursos informáticos.

Este servicio se puede utilizar por el primer año libre de costo (Figura 1.3), permitiendo implementar aplicaciones bajo ciertas restricciones en cuanto al procesamiento que el servidor ofrece [17].



Figura 1.3. Características de Instancia EC2

Servicio de Base de Datos:

A nivel de bases de datos uno de sus principales servicios es Amazon Relational Database Service (Amazon RDS).

- **Amazon Relational Database Service (Amazon RDS):** es un servicio web que facilita la configuración, el funcionamiento y el escalado de una base de datos relacional en AWS Cloud. Este servicio proporciona una capacidad rentable y de tamaño ajustable para una base de datos relacional estándar y se ocupa de las tareas de administración de bases de datos comunes y sus características son [18]:
 - Se pueden utilizar los sistemas de gestión de bases de datos como: MySQL, MariaDB, PostgreSQL, Oracle o Microsoft SQL Server.
 - Se puede realizar copias de seguridad automatizadas o crear manualmente una instantánea de copia de seguridad, estas copias de seguridad se pueden utilizar para restaurar una base de datos.
 - Administra las copias de seguridad, la aplicación de parches de software, la detección automática de errores y la recuperación.
 - Se puede escalar el tamaño del almacenamiento de acuerdo a las necesidades del cliente.

Este servicio permite almacenamiento por el primer año libre de costo (Figura 1.4), permitiendo implementar bases de datos bajo ciertas restricciones en cuanto al tamaño de disco duro [17].



Figura 1.4. Características de Instancia RDS

1.3.3.5 APLICACIONES MÓVILES

Una aplicación móvil es un tipo de software diseñado para ejecutarse en un dispositivo móvil como un teléfono inteligente, reloj inteligente o una Tablet. Las aplicaciones son generalmente pequeñas unidades de software individuales con funciones limitadas que proporcionan a los usuarios servicios similares a los que se accede en las PCs [19].

1.3.3.6 TIPOS DE APLICACIONES

A nivel de programación existen varias formas de desarrollar una aplicación, cada una de ellas tiene diferentes características y limitaciones, especialmente desde el punto de vista técnico [20].

- **Aplicaciones Nativas:**

Las aplicaciones móviles se desarrollan exclusivamente para un único sistema operativo móvil, por lo tanto, son nativas para una plataforma o dispositivo en particular. La aplicación creada para sistemas como iOS, Android, etc., no se puede utilizar en una plataforma que no sea la suya, en otras palabras, no se puede utilizar la aplicación de Android en iPhone [21].

La principal ventaja de las aplicaciones nativas es el alto rendimiento y garantizan una buena experiencia de usuario, ya que los desarrolladores utilizan la interfaz de usuario nativa del dispositivo y su meta es llegar a clientes objetivo [21].

- **Aplicaciones Híbridas:**

Las aplicaciones híbridas se desarrollan utilizando tecnologías web multiplataforma (por ejemplo, HTML5, CSS y JavaScript). Estas aplicaciones tienen una base de código único para todas las plataformas y dispositivos [21], permitiendo casi con un mismo código obtener diferentes aplicaciones, por ejemplo, para Android e iOS y distribuir las en cada una de sus tiendas [20].

Las desventajas de las aplicaciones híbridas es que carecen de rendimiento, velocidad y optimización general en comparación con las aplicaciones nativas. Además, hay ciertos problemas de diseño debido a la incapacidad de la aplicación para verse exactamente de la misma manera en dos o más plataformas distintas [21].

- **Aplicaciones Web:**

Las aplicaciones web utilizan un navegador web para ejecutarse y generalmente están desarrolladas en lenguajes como HTML5, JavaScript o CSS, lo cual permite programar de forma independiente al sistema operativo en el cual se usará la aplicación, estas aplicaciones pueden ser fácilmente utilizadas en diferentes plataformas sin mayores inconvenientes y sin necesidad de desarrollar un código diferente para cada caso particular [20].

Las aplicaciones web no necesitan instalarse, ya que se visualizan usando el navegador del teléfono como un sitio web normal y los usuarios pueden obtener acceso desde cualquier dispositivo siempre que haya conexión a Internet, el uso de aplicaciones web con una conexión deficiente daría lugar a una mala experiencia del usuario [21].

Al tratarse de aplicaciones que funcionan sobre la web no es necesario que el usuario reciba actualizaciones, ya que siempre va a estar viendo la última versión, a

diferencia de las aplicaciones nativas e híbridas que si requieren una actualización cada vez que se genera la última versión de la aplicación [20].

Para el presente proyecto, el tipo de aplicación móvil es nativa, ya que su desarrollo es específicamente para el sistema operativo Android.

1.3.3.7 ANDROID

Android es un sistema operativo móvil que se basa en una versión modificada de Linux [22] que fue diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes y Tablets [23].

El sistema operativo es abierto y gratuito debido a que la mayor parte del código de Android se lanzó bajo la licencia de código abierto, eso significa que cualquiera que quiera usar Android puede hacerlo descargando el código fuente completo de Android. Además, los proveedores (fabricantes de dispositivos móviles) pueden agregar sus propias extensiones propietarias y personalizar Android para diferenciar sus productos de los demás [24].

La principal ventaja de adoptar Android es que ofrece un enfoque unificado para el desarrollo de aplicaciones para dispositivos móviles, ya que los desarrolladores solo necesitan desarrollar para Android en general y sus aplicaciones deberían poder ejecutarse en numerosos dispositivos móviles de diferentes compañías, siempre y cuando los dispositivos funcionen con el sistema operativo Android [22].

Debido a que el sistema operativo Android es de código abierto y gratuito casi siempre es la primera opción entre los fabricantes de TV, consolas de videojuegos, sistemas de realidad aumentada, entre otros [22].

Arquitectura de Android

La arquitectura de Android (Figura 1.5) permite una amplia variedad de configuraciones y personalizaciones [25], se encuentra estructurada por las siguientes capas:

- **Kernel de Linux:** este es el kernel en el que está basado Android. Esta capa contiene todos los controladores de dispositivo de bajo nivel para los diversos componentes de hardware de un dispositivo Android [26].

- **Capa de abstracción de hardware (HAL):** el HAL consta de múltiples módulos de librería, cada uno de los cuales implementa una interfaz para un tipo específico de componente de hardware, como la cámara o el módulo bluetooth. Cuando el framework de una API hace una llamada para acceder al hardware del dispositivo, Android carga el módulo de librería para ese componente de hardware [27].
- **Librerías:** contienen el código que proporciona las características principales del sistema operativo Android y están escritas en C/C++. Por ejemplo, la librería SQLite proporciona soporte de base de datos para que una aplicación pueda usarla para el almacenamiento de datos y la librería WebKit proporciona funcionalidades para la navegación web [22].
- **Tiempo de ejecución de Android:** el tiempo de ejecución de Android se encuentra en la misma capa que las librerías y proporciona un conjunto de librerías principales que permiten a los desarrolladores escribir aplicaciones de Android utilizando el lenguaje de programación Java [22]. A partir de la versión Android 5.0 cada aplicación ejecuta sus procesos con sus propias instancias del tiempo de ejecución ART. ART fue diseñada para ejecutar varias máquinas virtuales específicamente para Android y optimizada para dispositivos móviles alimentados por batería con memoria y potencia de CPU limitadas [27].
- **Framework de la API de Java:** en esta capa se encuentran las librerías escritas en lenguaje Java para el desarrollo de aplicaciones, que permiten simplificar la reutilización de componentes del sistema, estas librerías permiten la utilización de sensores, localización, barra de notificaciones, etc. [26].
- **Aplicaciones:** en esta capa superior se encuentran las aplicaciones que se encuentran instaladas por defecto en el dispositivo Android (como contactos, cámara, SMS, calendario, correo electrónico, navegador de Internet, etc.), así como aquellas que el usuario vaya añadiendo posteriormente, ya sean las aplicaciones que se descarga e instala desde Google Play, de terceras empresas o de su propio desarrollo [22].

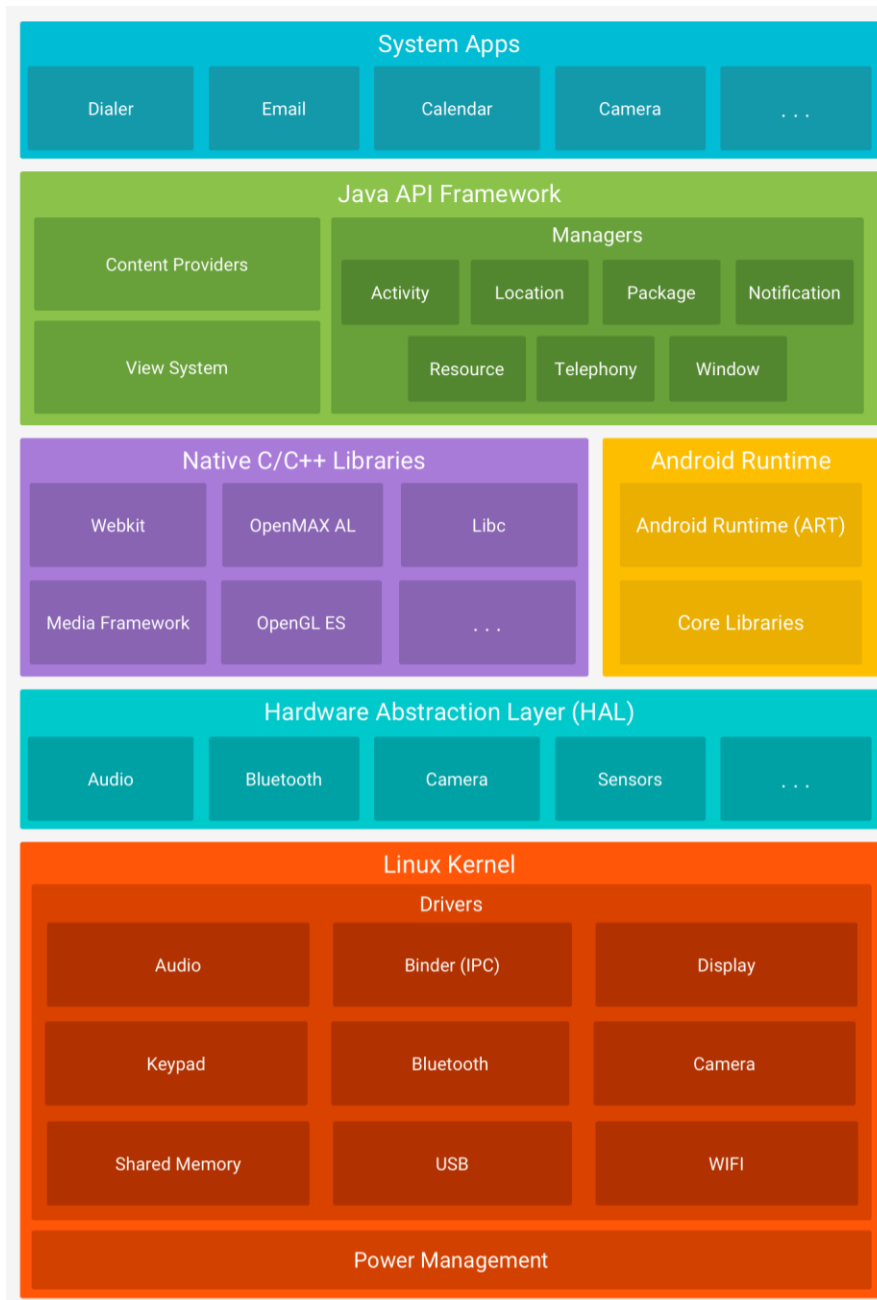


Figura 1.5. Arquitectura de Android

1.3.3.8 RETROFIT

Retrofit es un cliente HTTP de tipo REST para Java y Android, hace que sea relativamente fácil recuperar y cargar JSON (u otros datos estructurados) a través de un servicio web basado en REST [28].

En Retrofit se debe configurar un convertidor para la serialización de datos, por lo general para JSON se utiliza GSON, pero se puede agregar otros convertidores

personalizados para procesar XML u otros protocolos. Para trabajar con Retrofit básicamente se necesita [28]:

- Una interfaz que define la estructura las operaciones HTTP (Código 1.1).

```
public interface GitHubService {
    @GET("users/{user}/repos")
    Call<List<Repo>> listRepos(@Path("user") String user);
}
```

Código 1.1. Ejemplo de Interfaz con Método con Operación HTTP

- Un cliente que define la url base para poder acceder al servicio web (Código 1.2).

```
Retrofit retrofit = new Retrofit.Builder()
    .baseUrl("https://api.github.com/")
    .build();

GitHubService service = retrofit.create(GitHubService.class);
```

Código 1.2. Ejemplo de Cliente Retrofit

- Las llamadas con Call son para realizar una solicitud HTTP al servidor web remoto (Código 1.3).

```
Call<List<Repo>> repos = service.listRepos("octocat");
```

Código 1.3. Ejemplo de llamada Call

- Cada método de solicitud debe tener una anotación HTTP que proporcione el método de solicitud y la URL relativa, hay cuatro anotaciones incorporadas: GET, POST, PUT y DELETE (Código 1.4). La URL relativa del recurso se especifica en la anotación y el valor de retorno envuelve la respuesta en un objeto Call con el tipo del resultado esperado.

```
@GET("users/list")
```

Código 1.4. Ejemplo de Anotación GET de HTTP

- Se puede especificar un objeto para usarlo como un cuerpo de solicitud HTTP con la anotación @Body (Código 1.5).

```
@POST("users/new")
Call<User> createUser(@Body User user);
```

Código 1.5. Ejemplo de Método con Anotación POST

1.3.4 HERRAMIENTAS DE DESARROLLO

Se han empleado varias herramientas necesarias para la implementación de la base de datos, del servicio web y de la aplicación móvil.

1.3.4.1 MYSQL

MySQL es un sistema de gestión de bases de datos de código abierto que utiliza Lenguaje de Consulta Estructurado (SQL). MySQL es empleado en base de datos basadas en un modelo relacional, ya que utiliza múltiples tablas de datos para almacenar y administrar la información [29].

La codificación de la base de datos se realizó en MySQL Workbench, que es una herramienta visual de MySQL [30].

1.3.4.2 VISUAL STUDIO CODE

Visual Studio Code es un editor de código que está disponible para Windows, macOS y Linux. Tiene soporte para JavaScript, Node.js y diversas extensiones para otros lenguajes de programación [31].

1.3.4.3 NODE.JS

Node.js es un entorno de tiempo de ejecución JavaScript de código abierto y multiplataforma para el desarrollo de aplicaciones de servidor rápidas y escalables [32]. Una de las principales características de Node.js es emprender muchas tareas con poco consumo de recursos, imprescindible para el desarrollo de servicios de alta concurrencia, por lo que es ampliamente utilizada para el desarrollo de servicios web que devuelven datos en formato JSON, es decir servicios web REST. También se pueden construir aplicaciones de escritorio multiplataforma, compatibles con diversos sistemas operativos como Windows, Linux, Mac, etc. [33].

A continuación, se incluyen tres módulos muy importantes para el desarrollo del servicio web del proyecto:

- **Express:** este módulo proporciona una forma simple y potente de crear un servidor web [34].
- **Mysql:** su funcionalidad es conectar con la base de datos y realizar las consultas SQL [35].
- **Morgan:** este módulo permite visualizar el registro de solicitudes y errores en la consola [36].

1.3.4.4 ANDROID STUDIO

Android Studio es el Entorno de Desarrollo Integrado (IDE) para el desarrollo de aplicaciones de Android, proporciona una interfaz gráfica e incluye el Kit de Desarrollo de Software (SDK) con todas las librerías necesarias para el desarrollo de aplicaciones. Además, cuenta con un emulador para ejecutar aplicaciones sin la necesidad de un dispositivo real [22].

Kit de Desarrollo de Software (SDK)

El SDK de Android contiene todos los paquetes y herramientas necesarios para desarrollar una aplicación funcional de Android, incluye librerías, un emulador, documentación, código de muestra y tutoriales [37]. Los SDK llevan el nombre de la versión del sistema operativo Android al que corresponden, es decir, cada vez que Google lanza una nueva versión de Android, también se lanza un SDK correspondiente [22].

Se puede instalar varias versiones del SDK de Android y tener varias versiones del SDK disponibles permite desarrollar aplicaciones destinadas a dispositivos con diferentes versiones de Android [22]. Las plataformas de desarrollo que son compatibles con SDK incluyen sistemas operativos como Windows, Linux y Mac OS X [37].

Estructura del proyecto

Cada proyecto en Android Studio contiene uno o más archivos de código fuente y archivos de recursos y cuenta con las siguientes carpetas:

- **manifests:** contiene el archivo AndroidManifest.xml y cada aplicación debe contener este archivo, ya que describe información esencial sobre la aplicación, como [38]:

- El nombre del paquete de la aplicación.
 - Componentes, activities, servicios utilizados por la aplicación, etc.
 - Permisos que necesita para consumir recursos del sistema o de otras aplicaciones.
 - Características de hardware y software.
- **java:** contiene los archivos de código fuente de la aplicación en Java [38].
 - **res:** contiene todos los recursos que no son de código, es decir, recursos visuales de la aplicación, como el diseño de una interfaz de usuario (layout), imágenes de mapa de bits (drawable), colores, strings, fuentes de texto (font), etc. [39].

Activity

Una activity está conformada por dos partes: gráfica (un archivo XML) y lógica (un archivo Java). La parte gráfica (layout) contiene los elementos que conforman el diseño de la interfaz gráfica de la activity y es la que interactúa con el usuario. La parte lógica es la clase en la que se escribe el código que permite dar funcionalidad a la activity. Una aplicación puede contener una o más activities [40].

Toda activity debe tener por lo menos un método en la parte lógica, este método se denomina “onCreate”, que es responsable de la creación del activity y se utiliza para realizar todo tipo de inicializaciones, como la creación de la interfaz de usuario o la inicialización de estructuras de datos. Además, emplea el método setContentView, que es responsable de enlazar las dos partes del activity: gráfica y lógica [41].

1.3.5 KANBAN

Kanban es una metodología de desarrollo de software que tiene como principal objetivo gestionar como se completan las tareas de forma eficiente y con énfasis en la entrega justo a tiempo [42], permitiendo que los equipos de desarrollo de software puedan hacer coincidir la cantidad de trabajo con la capacidad del equipo, para lo cual requiere una comunicación en tiempo real. Esto brinda a los equipos opciones de planificación más flexibles, resultados más rápidos, un enfoque más claro y transparencia durante todo el ciclo de desarrollo [43].

Los elementos de trabajo se representan visualmente en un tablero Kanban, lo que permite a los miembros del equipo ver el estado de cada trabajo en cualquier momento [43].

1.3.5.1 TABLERO KANBAN

El trabajo de todos los equipos gira en torno a un tablero Kanban, una herramienta utilizada para visualizar el trabajo y optimizar el flujo del trabajo entre el equipo. Si bien los tableros físicos son populares entre algunos equipos, los tableros virtuales son una característica crucial en cualquier herramienta de desarrollo de software ágil por su trazabilidad, colaboración más fácil y accesibilidad desde múltiples ubicaciones [43].

Independientemente de si el tablero Kanban de un equipo es físico o digital, su función es garantizar que se visualice el trabajo del equipo, que su flujo de trabajo esté estandarizado y que todos los inconvenientes se identifiquen y resuelvan de inmediato [43].

Un tablero Kanban básico tiene un flujo de trabajo de tres pasos: hacer, en progreso y listo. Está basado en la transparencia total del trabajo y la capacidad de comunicación en tiempo real, por lo tanto, el tablero Kanban debe verse como la única fuente de verdad para el trabajo del equipo [43].

1.3.5.2 TARJETAS KANBAN

Para equipos voluminosos, cada elemento de trabajo se representa como una tarjeta separada en el tablero Kanban. El objetivo principal de representar el trabajo como una tarjeta en el tablero Kanban es permitir que los miembros del equipo sigan el progreso del trabajo a través de una manera altamente visual [43].

Las tarjetas Kanban presentan información crítica sobre un elemento de trabajo en particular, lo que permite a todos los miembros del equipo una visibilidad completa del estado de cada elemento de trabajo en un momento dado, quién es responsable de ese elemento de trabajo, una descripción del trabajo que se está realizando, cuánto tiempo se estima que llevará ese trabajo, etc. [43].

1.3.5.3 BENEFICIOS DE KANBAN

- **Flexibilidad de planificación:** un equipo solo se enfoca en el trabajo que está activamente en progreso, una vez que el equipo completa un elemento de trabajo, arrancan con el siguiente elemento de trabajo asignado [43].
- **Ciclos de tiempo cortos:** el ciclo de tiempo es la cantidad de tiempo que le toma a un equipo desde el momento en que comienza el trabajo hasta el momento en que

finaliza. Al optimizar el ciclo de tiempo, el equipo puede pronosticar la entrega del trabajo futuro [43].

Las habilidades compartidas significan que los miembros del equipo pueden optimizar aún más el ciclo de tiempo, ya que es responsabilidad de todo el equipo garantizar que el trabajo se desarrolle sin problemas durante todo el proceso [43].

- **Menos cuellos de botella:** cuantos más elementos de trabajo se encuentren en un momento dado, más se dificulta el camino hacia la finalización y un principio clave de Kanban es limitar la cantidad de trabajo en progreso. Los límites de trabajo en progreso permiten resaltar los cuellos de botella, la falta de enfoque o de habilidades de una o varias personas del equipo [43].
- **Entrega continua:** es la práctica de entregar trabajo a los clientes con frecuencia, incluso a diario o por hora, cuanto más rápido un equipo pueda ofrecer innovación al mercado, más competitivo será su producto en el mercado y los equipos de Kanban se centran exactamente en optimizar el flujo de trabajo a los clientes [43].

2 METODOLOGÍA

La metodología consta de dos componentes: el diseño del prototipo y la implementación de todos sus elementos. En la parte de diseño se especifica en un tablero Kanban las actividades fundamentales del proyecto, a continuación, se analizan todos los requerimientos que se deben cumplir y en base a ello se definen los módulos del aplicativo móvil, la arquitectura del prototipo y las tres capas que integran el diseño: capa de datos, capa de negocio y capa de presentación. En la parte de implementación, empleando los servicios en la nube de AWS y las herramientas de software especificadas en el Apartado 1.3.4, se codifican las tres capas: datos, negocio y presentación.

2.1 DISEÑO

Para el diseño del prototipo se define un tablero Kanban, en donde se detallan las actividades en las que se incluyen las de la etapa de diseño y que se encuentran en proceso de realización.

Luego de analizar las entrevistas realizadas al administrador, personal médico y personal dependiente de farmacia, se definen los requerimientos en dos tipos:

funcionales y no funcionales. A partir de estos requerimientos se especifican todos los módulos necesarios para el aplicativo móvil, de acuerdo al tipo de usuario.

A continuación, se define la arquitectura del prototipo y se realiza el diseño de cada capa, en la capa de datos se realiza el diagrama relacional de la base de datos, en la capa de negocio se realiza el diagrama de casos de uso de cada módulo, el diagrama de actividades de cada módulo y el diagrama de clases, y en la capa de presentación se especifican las vistas de cada módulo que componen la interfaz gráfica de la aplicación móvil.

2.1.1 TABLERO KANBAN

En la Tabla 2.1 se presenta el tablero Kanban inicial, el cual consta de tres columnas: Tareas Pendientes, Tareas en Proceso y Tareas Realizadas, en cada columna se especifican las tareas de acuerdo al estado en el que se encuentran.

Tabla 2.1. Tablero Kanban Inicial

Tareas Pendientes	Tareas en Proceso	Tareas Realizadas
Creación de una instancia de EC2 con Windows Server	Entrevistas a administrador, personal médico y dependiente de farmacia del centro de salud	Definición de conceptos teóricos
Creación de una instancia de RDS con MySql	Identificación de Requerimientos Funcionales	
Instalación de Node.js y Visual Studio Code en Windows Server	Identificación de Requerimientos No Funcionales	
Instalación de Android Studio	Definición de módulos de la aplicación móvil	
Codificación de Capa de Datos	Diseño de Capa de Datos	
Codificación de Capa de Negocio	Diseño de Capa de Negocio	
Codificación de Capa de Presentación	Diseño de Capa de Presentación	
Pruebas de funcionamiento del prototipo		
Encuestas de validación de funcionamiento a usuarios		

2.1.2 ANÁLISIS DE REQUERIMIENTOS

Para la obtención de requerimientos se identificaron tres actores: Administrador, Médico y Dependiente de Farmacia, a los cuales se entrevistó y, después de analizar los resultados de las entrevistas y en concordancia con el plan de trabajo de titulación aprobado, se obtuvo la información necesaria para determinar los requerimientos funcionales y no funcionales.

2.1.2.1 RESULTADOS DE ENTREVISTA A ADMINISTRADOR

En las Tablas 2.2, 2.3 y 2.4 se presentan los resultados de la entrevista realizada al administrador del centro de salud.

Tabla 2.2. Resultados de Entrevista a Administrador (Parte 1 de 3)

Pregunta	Resultado (%)
1. Los documentos para el registro farmacéutico de medicamentos se realizan de forma:	
a) Digital	
b) Manuscrita	100
c) Otra	
2. ¿Cuánto tiempo tarda en la realización de dichos documentos?	
a) 0-30 min.	
b) 30-60 min.	
c) 60-120 min.	100
d) más de 120 min.	
3. En caso de haber elegido la opción b) en la Pregunta 1. Piensa usted que la tarea de realizar manualmente los registros es:	
a) Muy Fácil	
b) Fácil	
c) Difícil	100
d) Muy Difícil	
4. En caso de haber elegido la opción b) en la Pregunta 1. Los registros farmacéuticos manuscritos de cada medicamento contienen una escritura de:	
a) Fácil Comprensión	
b) Mediana Comprensión	100
c) Difícil Comprensión	
5. Está usted de acuerdo con la digitalización de la información referente a los registros de medicamentos entrantes y salientes:	
a) Sí	100
b) No	
c) Talvez	
6. Usted considera que una aplicación móvil es una herramienta eficaz, ya que simplifica gran parte de los procesos a nivel profesional:	
a) Sí	100
b) No	
c) Talvez	

Tabla 2.3. Resultados de Entrevista a Administrador (Parte 2 de 3)

7. Usted considera útil que el registro farmacéutico de medicamentos se realice a través de una aplicación móvil, ya que dispondría de un fácil acceso a dichos datos:	
a) Sí	100
b) No	
c) Talvez	
8. ¿Qué Sistema Operativo tiene su teléfono móvil?	
a) iOS	
b) Android	100
c) Otro	
9. El centro de salud dispone de acceso a Internet a través de Wifi o Datos Móviles:	
a) Sí	100
b) No	
10. Está usted de acuerdo en ingresar sus datos personales en una aplicación móvil para registrarse como usuario:	
a) Sí	100
b) No	
Sí en la Pregunta 10 la respuesta fue a), responda las siguientes preguntas:	
11. ¿Cuáles datos está usted de acuerdo en emplear como sus credenciales de acceso a las funcionalidades de la aplicación móvil?	
a) Correo y contraseña	
b) Tipo de usuario, correo y contraseña	
c) Cedula y contraseña	
d) Tipo de usuario, cedula y contraseña	100
12. Está usted de acuerdo en visualizar sus datos personales registrados a través de la aplicación móvil:	
a) Sí	100
b) No	
13. Está usted de acuerdo en actualizar sus datos personales, actualizar su contraseña, desactivar o eliminar su registro, en caso de ser necesario:	
a) Sí	100
b) No	
14. Para crear el stock de medicamento. ¿Qué información considera usted de utilidad que se debería registrar de cada medicamento?	
a) Nombre de Medicamento y Unidades Totales	
b) Nombre de Medicamento, Concentración y Unidades Totales	
c) Tipo de Medicamento, Nombre de Medicamento y Unidades Totales	
d) Tipo de Medicamento, Nombre de Medicamento, Concentración y Unidades Totales	100
15. Para usted es útil visualizar en una lista todos los stocks de medicamentos registrados:	
a) Sí	100
b) No	

Tabla 2.4. Resultados de Entrevista a Administrador (Parte 3 de 3)

16. Está usted de acuerdo en eliminar un stock de medicamento de la lista, en caso de detectar un error en su registro:	
a) Sí	100
b) No	
17. Está usted de acuerdo en generar un historial farmacéutico de medicamento con los datos de entrada y salida:	
a) Sí	100
b) No	
c) Talvez	
En caso de haber elegido la opción a) en la Pregunta 17, responda las siguientes preguntas:	
18. ¿Qué datos usted considera necesarios para generar el historial farmacéutico de medicamento?	
a) Medicamento	
b) Tipo de Medicamento, Medicamento	
c) Tipo de Medicamento, Medicamento, Periodo de Tiempo	100
19. En caso de haber elegido la opción c) en la pregunta anterior. ¿Cuál tipo de período considera usted que debe disponer el historial de medicamento?	
a) Mensual	
b) Anual	
c) Total	
d) Mensual y Anual	
e) Mensual y Total	
f) Mensual, Anual y Total	100
20. Para usted qué nivel de utilidad representaría que la aplicación móvil pueda crear un documento digital en Pdf con la información del historial farmacéutico de medicamento:	
a) Muy útil	100
b) Útil	
c) Poco útil	
d) Nada útil	

2.1.2.2 RESULTADOS DE ENTREVISTAS A MÉDICOS

En las Tablas 2.5, 2.6 y 2.7 se presentan los resultados de las entrevistas realizadas a siete médicos del centro de salud.

Tabla 2.5. Resultados de Entrevistas a Médicos (Parte 1 de 3)

Pregunta	Resultado (%)
1. Para cada atención médica, el registro de datos del paciente y receta se realizan de forma:	
a) Digital	
b) Manuscrita	100
c) Otra	

Tabla 2.6. Resultados de Entrevistas a Médicos (Parte 2 de 3)

2. El registro de atenciones médicas sería más accesible a través de:	
a) Registros Físicos	28,57
b) Registros Digitales	71,43
3. Usted considera que una aplicación móvil es una herramienta eficaz, ya que simplifica gran parte de los procesos a nivel profesional:	
a) Sí	85,71
b) No	14,29
c) Talvez	0
4. Usted considera útil registrar los datos del paciente y medicamentos de receta de cada atención médica a través de una aplicación móvil:	
a) Sí	71,42
b) No	14,29
c) Talvez	14,29
5. Usted considera que, a través de una aplicación móvil facilitaría el acceso a información de las atenciones médicas registradas con anterioridad:	
a) Sí	85,71
b) No	14,29
c) Talvez	0
6. ¿Qué Sistema Operativo tiene su teléfono móvil?	
a) iOS	28,57
b) Android	71,43
c) Otro	0
7. Está usted de acuerdo en ingresar sus datos personales en una aplicación móvil para registrarse como usuario:	
a) Sí	85,71
b) No	14,29
Sí en la Pregunta 7 la respuesta fue a), responda las siguientes preguntas:	
8. ¿Cuáles datos está usted de acuerdo en emplear como sus credenciales de acceso a las funcionalidades de la aplicación móvil?	
a) Correo y contraseña	16,66
b) Tipo de Usuario, correo y contraseña	16,66
c) Cedula y contraseña	16,66
d) Tipo de Usuario, cedula y contraseña	50
9. Está usted de acuerdo en visualizar sus datos personales registrados a través de la aplicación móvil:	
a) Sí	66,67
b) No	0
c) Talvez	33,33
10. Está usted de acuerdo en actualizar sus datos personales, actualizar su contraseña, desactivar o eliminar su registro, en caso de ser necesario:	
a) Sí	66,66
b) No	16,66
c) Talvez	16,66

Tabla 2.7. Resultados de Entrevistas a Médicos (Parte 3 de 3)

11. Para crear una atención médica. ¿Qué información personal del paciente considera usted de utilidad que se debería registrar?	
a) Apellidos y Nombres	0
b) Apellidos, Nombres y Sexo	16,67
c) Cédula, Apellidos y Nombres	33,33
d) Cédula, Apellidos, Nombres y Sexo	50
12. Al elegir un medicamento para añadirse a la receta. Está usted de acuerdo en incluir dicho medicamento en la receta, sólo si existe el stock requerido del mismo:	
a) Sí	100
b) No	0
c) Talvez	0
13. Está usted de acuerdo en visualizar los detalles de las atenciones médicas que ha registrado previamente:	
a) Sí	100
b) No	0
c) Talvez	0
14. En una atención médica registrada previamente, está usted de acuerdo en reemplazar un medicamento de la receta por otro similar, en caso de ser insuficiente el stock del mismo:	
a) Sí	83,33
b) No	0
c) Talvez	16,67
15. En una atención médica registrada previamente, está usted de acuerdo en eliminar un medicamento de la receta, en caso de ser insuficiente el stock del mismo y no exista en stock un medicamento similar:	
a) Sí	83,33
b) No	0
c) Talvez	16,67

2.1.2.3 RESULTADOS DE ENTREVISTAS A DEPENDIENTES DE FARMACIA

En las tablas 2.8, 2.9 y 2.10 se presentan los resultados de las entrevistas realizadas a cuatro dependientes de farmacia médicos del centro de salud.

Tabla 2.8. Resultados de Entrevistas a Dependientes de Farmacia (Parte 1 de 3)

Pregunta	Resultado (%)
1. Las recetas emitidas por los médicos del centro de salud tienen escritura clara y legible:	
a) Pocas veces	50
b) A veces	25
c) Muchas veces	25

Tabla 2.9. Resultados de Entrevistas a Dependientes de Farmacia (Parte 2 de 3)

2. Los pacientes presentan las recetas recibidas durante una atención médica para reclamar sus medicamentos:	
a) Pocas veces	50
b) A veces	50
c) Muchas veces	0
3. El registro de entrega de los medicamentos de receta se realiza de forma:	
a) Digital	0
b) Manuscrita	100
c) Otra	0
4. La búsqueda de los datos de registros de entrega de medicamentos realizados con anterioridad es:	
a) Fácil	
b) Medianamente Difícil	75
c) Muy Difícil	25
5. Usted considera que una aplicación móvil es una herramienta eficaz, ya que simplifica gran parte de los procesos a nivel profesional:	
a) Sí	75
b) No	0
c) Talvez	25
6. Usted considera que una aplicación móvil facilitaría registrar la entrega de medicamentos de receta de cada atención médica, sin la necesidad de documentos manuscritos:	
a) Sí	75
b) No	0
c) Talvez	25
7. Usted considera que una aplicación móvil facilitaría el acceso a la información sobre las atenciones médicas cuyos medicamentos de receta constan con registro de entrega:	
a) Sí	75
b) No	0
c) Talvez	25
8. ¿Qué Sistema Operativo tiene su teléfono móvil?	
a) iOS	25
b) Android	75
c) Otro	0
9. Está usted de acuerdo en ingresar sus datos personales en una aplicación móvil para registrarse como usuario:	
a) Sí	75
b) No	25
Sí en la Pregunta 9 la respuesta fue a), responda las siguientes preguntas:	
10. ¿Cuáles datos está usted de acuerdo en emplear como sus credenciales de acceso a las funcionalidades de la aplicación móvil?	
a) Correo y contraseña	0
b) Tipo de Usuario, correo y contraseña	33,33
c) Cedula y contraseña	0
d) Tipo de Usuario, cedula y contraseña	66,67

Tabla 2.10. Resultados de Entrevistas a Dependientes de Farmacia (Parte 3 de 3)

11. Está usted de acuerdo en visualizar sus datos personales registrados a través de la aplicación móvil:	
a) Sí	66,67
b) No	0
c) Talvez	33,33
12. Está usted de acuerdo en actualizar sus datos personales, actualizar su contraseña, desactivar o eliminar su registro, en caso de ser necesario:	
a) Sí	100
b) No	0
c) Talvez	0
13. Para la entrega de los medicamentos de cada receta. Está usted de acuerdo en buscar los datos de la atención médica por:	
a) Paciente	66,67
b) Especialidad	33,33
c) Médico	0
14. Está usted de acuerdo en verificar la existencia del stock requerido de cada medicamento de la receta, en el momento de registrar su entrega:	
a) Sí	100
b) No	0
c) Talvez	0
15. Está usted de acuerdo en actualizar la atención médica sí se ha registrado la entrega de todos los medicamentos de la receta a un estado de finalizado:	
a) Sí	100
b) No	0
c) Talvez	0
16. Está usted de acuerdo en visualizar las atenciones médicas que constan con registro de entrega de todos los medicamentos de receta:	
a) Sí	100
b) No	0
c) Talvez	0

2.1.2.4 REQUERIMIENTOS FUNCIONALES

Los requerimientos funcionales describen el funcionamiento del sistema y se obtuvieron en base a los resultados de las entrevistas realizadas a los tres tipos de usuarios de la aplicación móvil: Administrador, Médico y Dependiente de Farmacia, y de acuerdo al plan de trabajo de titulación aprobado. En la Tabla 2.11 se describe los requerimientos funcionales identificados con la nomenclatura RF para cada requerimiento funcional.

Tabla 2.11. Requerimientos Funcionales

CODIGO RF	DESCRIPCIÓN
RF01	Crear Nuevo Usuario
RF02	Ingresar Credenciales de Usuario
RF03	Visualizar Datos de Usuario
RF04	Actualizar Datos de Usuario
RF05	Actualizar Contraseña de Usuario
RF06	Desactivar Usuario
RF07	Eliminar Usuario
RF08	Ingresar Datos de un Medicamento
RF09	Crear Stock de Medicamento
RF10	Visualizar Medicamentos en Stock
RF11	Eliminar Medicamento en Stock
RF12	Seleccionar un Medicamento para Historial
RF13	Escoger Tipo de Historial
RF14	Generar Historial de Medicamento
RF15	Crear Pdf con Historial de Medicamento
RF16	Ingresar Datos de Paciente
RF17	Seleccionar Medicamento de Receta
RF18	Crear Atención Médica
RF19	Visualizar Atención Médica Registrada
RF20	Reemplazar Medicamento de Receta por otro
RF21	Eliminar Medicamento de Receta
RF22	Buscar Paciente Registrado
RF23	Visualizar Atención Médica Pendiente de Entrega de Medicamentos de Paciente
RF24	Registrar Entrega de Medicamento de Receta
RF25	Actualizar Atención Médica
RF26	Visualizar Atención Médica Finalizada

2.1.2.5 REQUERIMIENTOS NO FUNCIONALES

Estos requerimientos describen las cualidades del sistema y no funciones a realizarse. En la Tabla 2.12 se describen los requerimientos no funcionales, empleando la nomenclatura RNF para cada requerimiento no funcional.

Tabla 2.12. Requerimientos No Funcionales

CÓDIGO	DESCRIPCIÓN
RNF01	El diseño de la aplicación móvil debe permitir su funcionamiento sólo en dispositivos Android.
RNF02	Para la utilización de la aplicación móvil por parte de los usuarios se requiere una conexión a Internet.
RNF03	A medida que aumentan el número de usuarios de la aplicación móvil, el prototipo debe permitir la escalabilidad de los recursos de procesamiento, almacenamiento y memoria.
RNF04	El prototipo debe tener una alta disponibilidad de tiempo de usabilidad.

2.1.3 MÓDULOS DEL PROTOTIPO

En base a los requerimientos descritos se determinaron tres tipos de usuarios de la aplicación móvil: ADMINISTRADOR, MÉDICO y DEPENDIENTE DE FARMACIA, y se definieron los módulos de tipo común: utilizados por todos los usuarios, y módulos específicos: cada usuario tiene acceso a una funcionalidad específica.

MÓDULOS COMUNES

- **Registro de Usuario:** el usuario registra sus datos personales, para ello debe ingresar cédula, apellidos, nombres, sexo, seleccionar el tipo de usuario entre las tres opciones: Administrador, Médico o Dependiente de Farmacia, en caso de elegir Médico, debe seleccionar la especialidad médica. Finalmente, debe ingresar una contraseña.
Existe un único Administrador activo y es el encargado de validar los registros de los usuarios Médico y Dependiente de Farmacia.
- **Inicio de Sesión de Usuario:** el usuario debe seleccionar el tipo de usuario e ingresar sus credenciales previamente registradas: cédula y contraseña, para acceder al menú principal de acuerdo al tipo de usuario correspondiente.

- **Datos Personales de Usuario:** se visualizan los datos personales del usuario y dispone de varias opciones para ser realizadas por el usuario: actualizar datos personales, actualizar contraseña, desactivar cuenta o eliminar cuenta.

MÓDULOS DE USUARIO ADMINISTRADOR

- **Nuevo Stock de Medicamento:** el administrador registra un nuevo stock de un medicamento específico, para ello debe ingresar un nuevo tipo de medicamento o seleccionar un tipo registrado. Si ingresa un nuevo tipo, debe ingresar un nuevo medicamento, para ello debe especificar el nombre y la concentración. Si selecciona un tipo registrado, debe ingresar un nuevo medicamento o seleccionar un medicamento registrado. Finalmente debe ingresar la cantidad en unidades del medicamento.
- **Lista de Medicamentos en Stock:** el administrador visualiza la lista con los detalles de cada uno de los medicamentos en stock que previamente ha registrado, y en caso de existencia de errores en alguno de los registros puede seleccionarlo para eliminarlo.
- **Historial Farmacéutico de Medicamento:** el administrador genera y visualiza un historial farmacéutico de un medicamento específico, en el cual se detallan los datos de entrada y salida del mismo, para ello debe seleccionar el tipo de medicamento, el medicamento y escoger el tipo de historial: total, anual o mensual. Además, puede crear un archivo en formato Pdf con dichos datos.

MÓDULOS DE USUARIO MÉDICO

- **Nueva Atención Médica:** el médico registra la atención médica que ha realizado en la que se incluye los datos personales del paciente: apellidos, nombres, cédula y sexo, en caso de haber sido registrado previamente, se buscan los datos del paciente en la base de datos. El siguiente paso es añadir uno o varios medicamentos a la receta y para cada uno primero debe seleccionar el tipo de medicamento, el medicamento e ingresar el número de unidades requeridas, y se verifica la existencia de las unidades requeridas del medicamento en stock. Si existe, se agrega el medicamento a la receta, caso contrario, no se agrega y puede sustituirlo por un medicamento equivalente. La atención médica se registra con un estado "Pendiente Entrega", es decir, requiere registro de entrega de medicamentos.

- **Atenciones Médicas Registradas:** el médico visualiza todas las atenciones médicas que ha registrado, en cada una de ellas consta el nombre del paciente, la fecha de la atención médica y el estado en que se encuentra en ese momento: “Pendiente Entrega”, “Parcialmente Entregado” o “Entregado Completo”. Al seleccionar una atención médica se detalla la lista de medicamentos de la receta. Cuando una atención médica tiene estado “Pendiente Entrega” o “Parcialmente Entregado”, el médico puede actualizar o eliminar uno o varios medicamentos de dicha receta. Cuando una atención médica tiene estado “Entregado Completo”, el médico no puede modificar ningún medicamento de la receta.

MÓDULOS DE USUARIO DEPENDIENTE DE FARMACIA

- **Atenciones Médicas Por Entregar Medicamentos:** el dependiente de farmacia debe ingresar la cédula de un paciente para visualizar las atenciones médicas recibidas por dicho paciente y que consten con estado “Pendiente Entrega” o “Parcialmente Entregado”. Cada atención médica contiene la fecha de atención médica, los datos del paciente, médico y la lista de medicamentos de la receta. El dependiente debe registrar la entrega de cada medicamento de la receta que no tenga registro de entrega, para ello verifica la existencia de la cantidad de unidades requeridas del medicamento en stock. En caso de existir, el sistema registra la fecha de entrega, caso contrario, no se registra la entrega de dicho medicamento. Una vez realizado este proceso, el estado de la atención médica se actualiza a “Entregado Completo”, “Parcialmente Entregado” o “Pendiente Entrega”.
Estados de la atención médica:
 - Entregado Completo: El dependiente ha registrado la entrega de todos los medicamentos de la receta.
 - Parcialmente Entregado: El dependiente ha registrado parcialmente la entrega de los medicamentos de la receta por falta de stock de uno o varios medicamentos.
 - Pendiente Entrega: El dependiente no ha registrado la entrega de ninguno de los medicamentos de la receta, es decir, mantiene el estado inicial.
- **Atenciones Médicas Finalizadas:** el dependiente de farmacia visualiza la lista de atenciones médicas que ya constan con estado “Entregado Completo”, es decir, se ha registrado la entrega de todos los medicamentos de la receta. Cada atención médica contiene los datos del paciente, médico y la lista de medicamentos de la receta con la fecha de entrega en cada medicamento.

En la Tabla 2.13 se presenta el requerimiento funcional para el módulo Registro de Usuario y el tiempo estimado de ejecución.

Tabla 2.13. Estimación RF Módulo Registro de Usuario

MÓDULO	REQUERIMIENTO FUNCIONAL	HORAS ESTIMADAS
Registro de Usuario	RF01	25
	TOTAL	25

En la Tabla 2.14 se presenta el requerimiento funcional para el módulo Inicio de Sesión de Usuario y el tiempo estimado de ejecución.

Tabla 2.14. Estimación RF Módulo Inicio de Sesión de Usuario

MÓDULO	REQUERIMIENTO FUNCIONAL	HORAS ESTIMADAS
Inicio de Sesión de Usuario	RF02	20
	TOTAL	20

En la Tabla 2.15 se presentan los requerimientos funcionales para el módulo Datos Personales de Usuario y el tiempo estimado de ejecución.

Tabla 2.15. Estimación RF Módulo Datos Personales de Usuario

MÓDULO	REQUERIMIENTO FUNCIONAL	HORAS ESTIMADAS
Datos Personales de Usuario	RF03	10
	RF04	10
	RF05	5
	RF06	5
	RF07	5
	TOTAL	35

En la Tabla 2.16 se presentan los requerimientos funcionales para el módulo Nuevo Stock de Medicamento y el tiempo estimado de ejecución.

Tabla 2.16. Estimación RF Módulo Nuevo Stock de Medicamento

MÓDULO	REQUERIMIENTO FUNCIONAL	HORAS ESTIMADAS
Nuevo Stock de Medicamento	RF08	15
	RF09	20
	TOTAL	35

En la Tabla 2.17 se presentan los requerimientos funcionales para el módulo Lista de Medicamentos en Stock y el tiempo estimado de ejecución.

Tabla 2.17. Estimación RF Módulo Lista de Medicamentos en Stock

MÓDULO	REQUERIMIENTO FUNCIONAL	HORAS ESTIMADAS
Lista de Medicamentos en Stock	RF10	15
	RF11	5
	TOTAL	20

En la Tabla 2.18 se presentan los requerimientos funcionales para el módulo Historial Farmacéutico de Medicamento y el tiempo estimado de ejecución.

Tabla 2.18. Estimación RF Módulo Historial Farmacéutico de Medicamento

MÓDULO	REQUERIMIENTO FUNCIONAL	HORAS ESTIMADAS
Historial Farmacéutico de Medicamento	RF12	10
	RF13	10
	RF14	10
	RF15	20
	TOTAL	50

En la Tabla 2.19 se presentan los requerimientos funcionales para el módulo Nueva Atención Médica y el tiempo estimado de ejecución.

Tabla 2.19. Estimación RF Módulo Nueva Atención Médica

MÓDULO	REQUERIMIENTO FUNCIONAL	HORAS ESTIMADAS
Nueva Atención Médica	RF16	15
	RF17	15
	RF18	15
	TOTAL	45

En la Tabla 2.20 se presentan los requerimientos funcionales para el módulo Atenciones Médicas Registradas y el tiempo estimado de ejecución.

Tabla 2.20. Estimación RF Módulo Atenciones Médicas Registradas

MÓDULO	REQUERIMIENTO FUNCIONAL	HORAS ESTIMADAS
Atenciones Médicas Registradas	RF19	10
	RF20	10
	RF21	5
	TOTAL	25

En la Tabla 2.21 se presentan los requerimientos funcionales para el módulo Atenciones Médicas Por Entregar Medicamentos y el tiempo estimado de ejecución.

Tabla 2.21. Estimación RF Módulo Atenciones Médicas Por Entregar Medicamentos

MÓDULO	REQUERIMIENTO FUNCIONAL	HORAS ESTIMADAS
Atenciones Médicas Por Entregar Medicamentos	RF22	10
	RF23	10
	RF24	15
	RF25	15
	TOTAL	50

En la Tabla 2.22 se presenta el requerimiento funcional para el módulo Atenciones Médicas Finalizadas y el tiempo estimado de ejecución.

Tabla 2.22. Estimación RF Módulo Atenciones Médicas Finalizadas

MÓDULO	REQUERIMIENTO FUNCIONAL	HORAS ESTIMADAS
Atenciones Médicas Finalizadas	RF26	10
	TOTAL	10

2.1.4 ARQUITECTURA DEL PROTOTIPO

La arquitectura (Figura 2.1) está basada en tres capas: capa de presentación, capa de negocio y capa de datos. La capa de presentación está conformada por la interfaz gráfica de la aplicación móvil en Android y es utilizada por el usuario final para realizar peticiones y recibir respuestas en formato JSON desde el servicio web. La capa de negocio se encarga de enlazar la capa de presentación con la capa de datos, está conformada por el servicio web REST codificado en JavaScript, y cada vez que recibe una petición del usuario final, se encarga de hacer las consultas SQL en la base de datos y una vez recibe los datos requeridos, envía la respuesta al usuario final. La capa de datos es donde residen los datos y está conformada por la base de datos codificada en MySQL y se encarga de procesar las consultas SQL del servicio web.

2.1.5 DISEÑO DE CAPA DE DATOS

La capa de datos está representada por el modelo relacional.

2.1.5.1 DIAGRAMA RELACIONAL

El diagrama relacional (Figura 2.2) permite modelar la base de datos, está compuesto por tablas donde se almacena la información y la relación existente entre dichas tablas [44], en este caso se han identificado diez tablas para la base de datos.

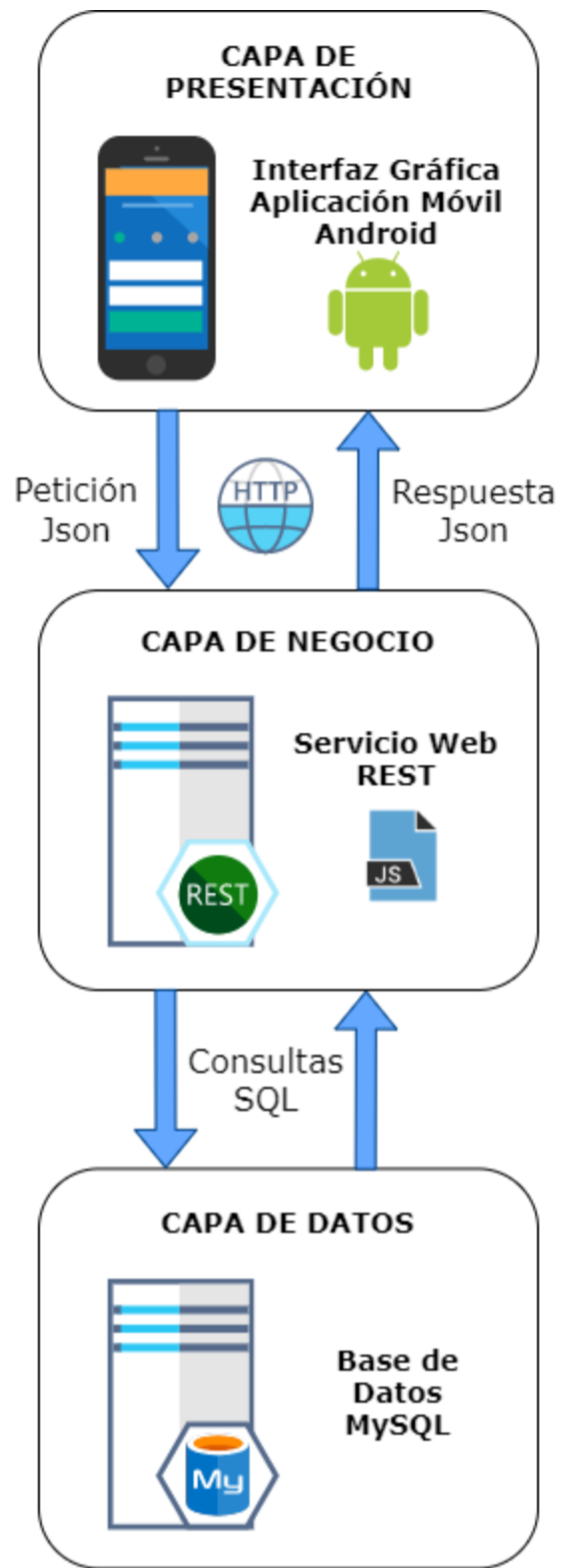


Figura 2.1. Arquitectura del Prototipo

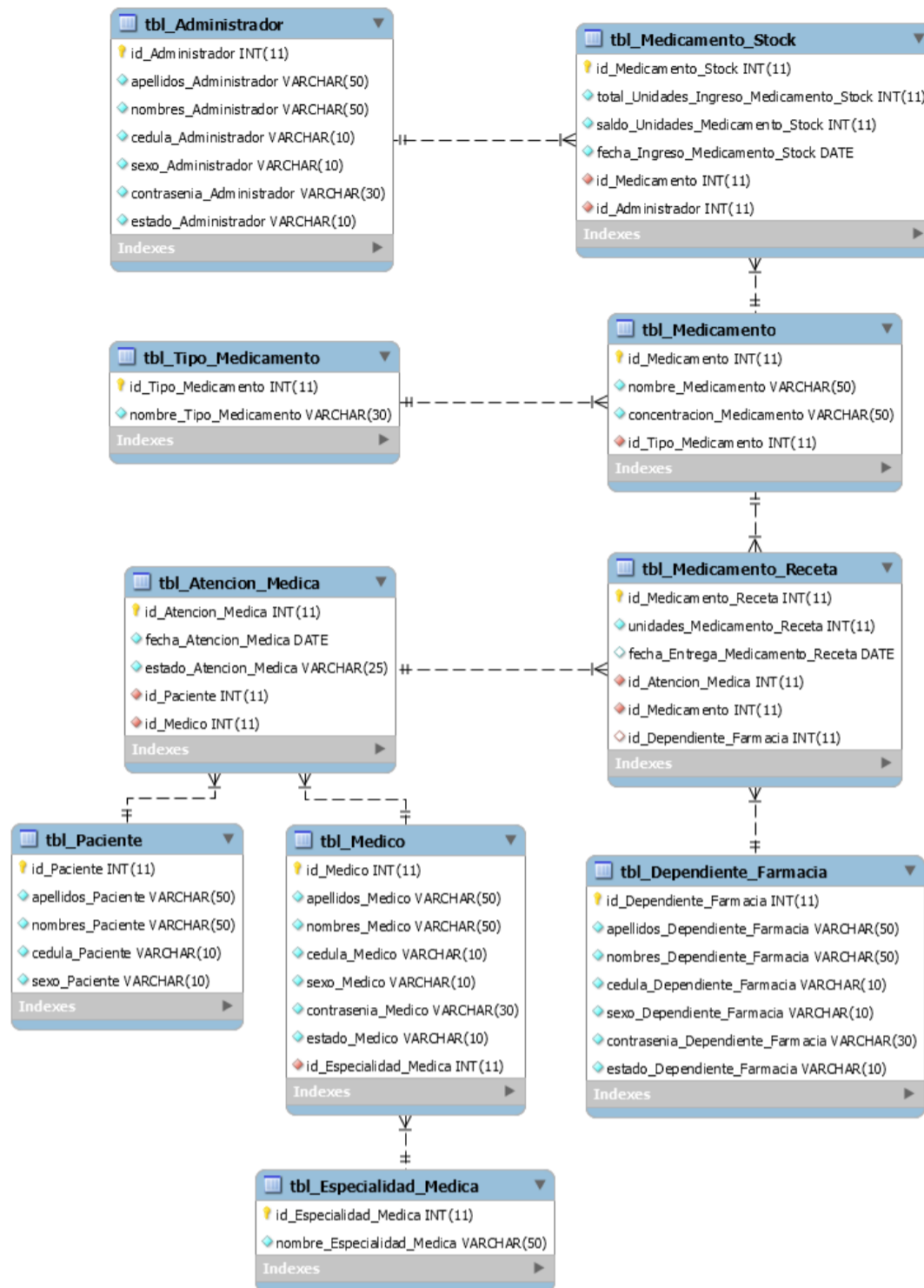


Figura 2.2. Diagrama Relacional

- **Tabla tbl_Administrador:** contiene los datos del usuario Administrador, el campo id_Administrador es la llave primaria.
- **Tabla tbl_Medicamento_Stock:** contiene los datos de cada stock de medicamentos registrado por el administrador, el campo id_Medicamento_Stock es la llave primaria

y los campos id_Administrador e id_Medicamento son las llaves foráneas, para establecer la relación con las tablas tbl_Administrador y tbl_Medicamento.

- **Tabla tbl_Medicamento:** contiene los datos de cada medicamento, el campo id_Medicamento es la llave primaria y el campo id_Tipo_Medicamento es la llave foránea, para establecer la relación con la tabla tbl_Tipo_Medicamento.
- **Tabla tbl_Tipo_Medicamento:** contiene los datos de cada tipo de medicamento, el campo id_Tipo_Medicamento es la llave primaria.
- **Tabla tbl_Medicamento_Receta:** contiene los datos de los medicamentos de cada atención médica, el campo id_Medicamento_Receta es la llave primaria y los campos id_Atencion_Medica, id_Medicamento e id_Dependiente_Farmacia son las llaves foráneas, para establecer la relación con las tablas tbl_Atencion_Medica, tbl_Medicamento y tbl_Dependiente_Farmacia.
- **Tabla tbl_Dependiente_Farmacia:** contiene los datos de cada usuario Dependiente de Farmacia, el campo id_Dependiente_Farmacia es la llave primaria.
- **Tabla tbl_Atencion_Medica:** contiene los datos de cada atención médica, el campo id_Atencion_Medica es la llave primaria y los campos id_Medico e id_Paciente son las llaves foráneas, para establecer la relación con las tablas tbl_Medico y tbl_Paciente.
- **Tabla tbl_Medico:** contiene los datos de cada usuario Médico, el campo id_Medico es la llave primaria y el campo id_Especialidad_Medica es la llave foránea, para establecer la relación con la tabla tbl_Especialidad_Medica.
- **Tabla tbl_Especialidad_Medica:** contiene los datos de cada especialidad médica, el campo id_Especialidad_Medica es la llave primaria.
- **Tabla tbl_Paciente:** contiene los datos de cada paciente, el campo id_Paciente es la llave primaria.

2.1.6 DISEÑO DE CAPA DE NEGOCIO

En esta capa se encuentra la lógica de los procesos que deben cumplirse para la implementación del prototipo y está compuesta por los diagramas de casos de uso, los diagramas de actividades y el diagrama de clases.

2.1.6.1 DIAGRAMA DE CASOS DE USO

El diagrama de casos de uso permite especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios, es decir, muestra la relación entre los actores y los casos de uso en un sistema [45].

Para la elaboración de los diagramas de casos de uso, se han identificado los casos de uso acorde a cada uno de los requerimientos funcionales especificados en el apartado 2.1.2.4. En la Tabla 2.23 se presentan los casos de uso definidos para los módulos más representativos, es decir, los módulos específicos de cada usuario y se identifican con el código CU.

Tabla 2.23. Casos de Uso de Cada Módulo

MÓDULO	CODIGO RF	CASO DE USO	DESCRIPCIÓN
Nuevo Stock de Medicamento	RF08	CU01	Ingresar Datos de un Medicamento
	RF09	CU02	Crear Stock de Medicamento
Lista de Medicamentos en Stock	RF10	CU03	Visualizar Medicamentos en Stock
	RF11	CU04	Eliminar Medicamento en Stock
Historial Farmacéutico de Medicamento	RF12	CU05	Seleccionar un Medicamento para Historial
	RF13	CU06	Escoger Tipo de Historial
	RF14	CU07	Generar Historial de Medicamento
	RF15	CU08	Crear Pdf con Historial de Medicamento
Nueva Atención Médica	RF16	CU09	Ingresar Datos de Paciente
	RF17	CU10	Seleccionar Medicamento de Receta
	RF18	CU11	Crear Atención Médica
Atenciones Médicas Registradas	RF19	CU12	Visualizar Atención Médica Registrada
	RF20	CU13	Reemplazar Medicamento de Receta por otro
	RF21	CU14	Eliminar Medicamento de Receta
Atenciones Médicas Por Entregar Medicamentos	RF22	CU15	Buscar Paciente Registrado
	RF23	CU16	Visualizar Atención Médica Pendiente de Entrega de Medicamentos de Paciente
	RF24	CU17	Registrar Entrega de Medicamento de Receta
	RF25	CU18	Actualizar Atención Médica
Atenciones Médicas Finalizadas	RF26	CU19	Visualizar Atención Médica Finalizada

ADMINISTRADOR:

- **Nuevo Stock de Medicamento:**

En este módulo el administrador ingresa los datos de un medicamento y crea el stock de dicho medicamento (Figura 2.3).

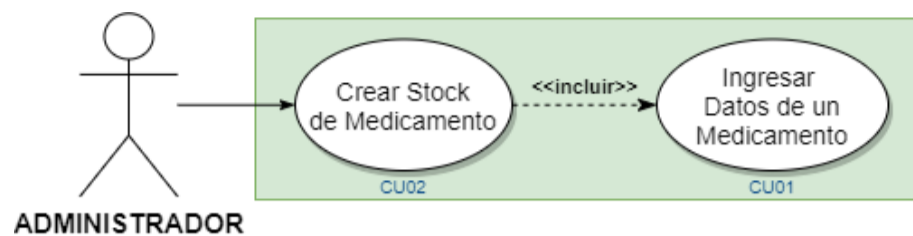


Figura 2.3. Diagrama de Casos de Uso: Nuevo Stock de Medicamento

- **Lista de Medicamentos en Stock:**

En este módulo el administrador visualiza todos los medicamentos en stocks registrados previamente y puede eliminar alguno si lo requiere (Figura 2.4).

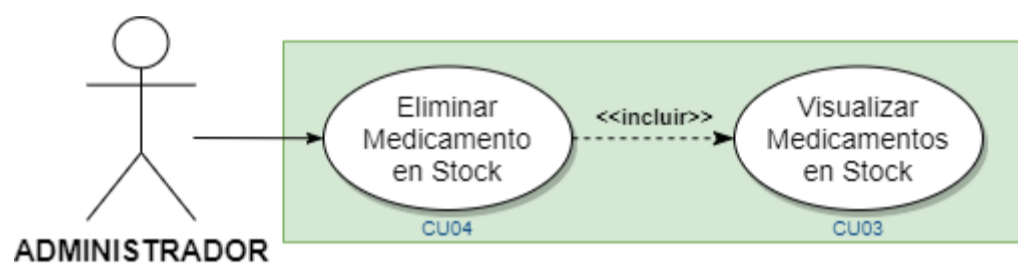


Figura 2.4. Diagrama de Casos de Uso: Lista de Medicamentos en Stock

- **Historial Farmacéutico de Medicamento:**

En este módulo el administrador selecciona un medicamento para el historial, escoge el tipo de historial, genera el historial de medicamento y crea el Pdf con el historial del medicamento (Figura 2.5).

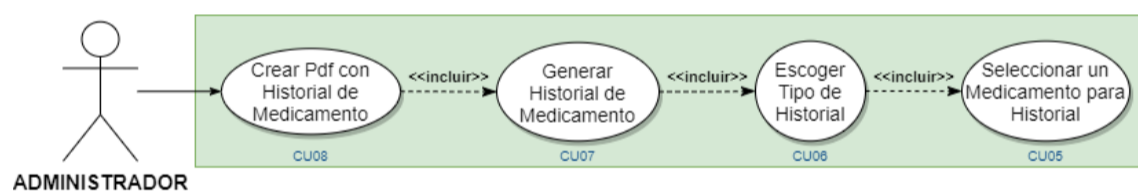


Figura 2.5. Diagrama de Casos de Uso: Historial Farmacéutico de Medicamento

MÉDICO:

- **Nueva Atención Médica:**

En este módulo el médico ingresa los datos de un paciente, selecciona cada medicamento de la receta y crea la atención médica (Figura 2.6).

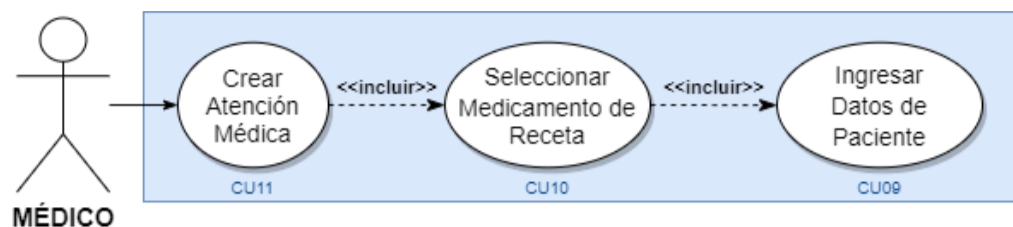


Figura 2.6. Diagrama de Casos de Uso: Nueva Atención Médica

- **Atenciones Médicas Registradas:**

En este módulo el médico visualiza todas las atenciones médicas que ha registrado previamente, de una atención médica específica, puede reemplazar un medicamento de la receta por otro o eliminar uno o varios medicamentos de la receta si lo requiere (Figura 2.7).

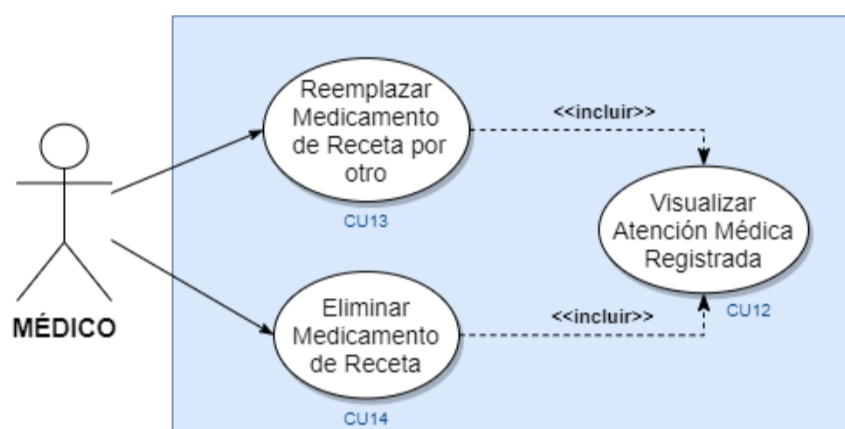


Figura 2.7. Diagrama de Casos de Uso: Atenciones Médicas Registradas

DEPENDIENTE DE FARMACIA:

- **Atenciones Médicas Por Entregar Medicamentos:**

En este módulo el dependiente de farmacia realiza la búsqueda de un paciente y visualiza las atenciones médicas recibidas por dicho paciente con entrega pendiente de medicamentos. A continuación, registra la entrega de cada medicamento de la receta y actualiza la atención médica (Figura 2.8).

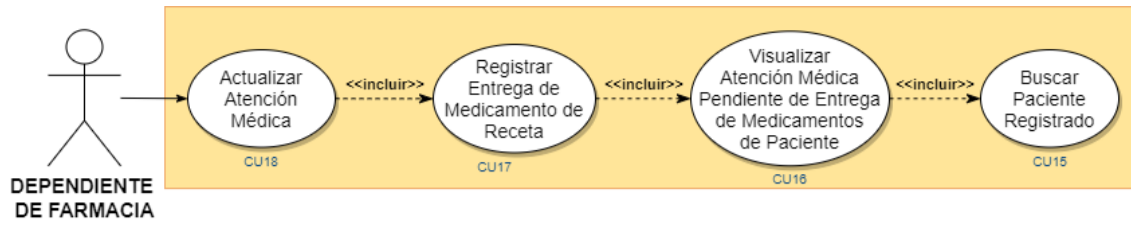


Figura 2.8. Diagrama de Casos de Uso: Atenciones Médicas Por Entregar Medicamentos

- **Atenciones Médicas Finalizadas:**

En este módulo el dependiente de farmacia visualiza todas las atenciones médicas finalizadas (Figura 2.9).

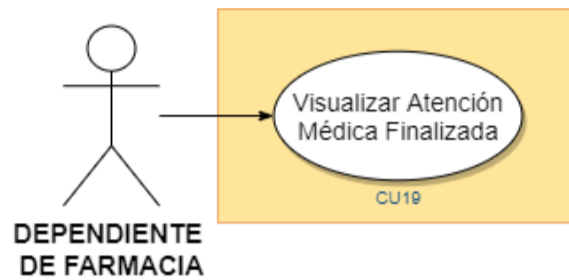


Figura 2.9. Diagrama de Casos de Uso: Atenciones Médicas Finalizadas

2.1.6.2 DIAGRAMA DE ACTIVIDADES

Los diagramas de actividades permiten establecer la secuencia de acción del sistema desde un punto inicial hasta un punto final, es decir, los pasos involucrados en la ejecución de un caso de uso [46]. A continuación, se describen los diagramas de actividades de los casos de uso anteriores:

- **Nuevo Stock de Medicamento:**

Para iniciar el proceso, el administrador tiene dos opciones a escoger: ingresar un nuevo tipo de medicamento o seleccionar un tipo de medicamento registrado. Si escoge la primera opción, debe ingresar los datos de un nuevo medicamento, y en la segunda opción, debe seleccionar un medicamento registrado o ingresar los datos de un nuevo medicamento. A continuación, debe ingresar las unidades totales de dicho medicamento y crear el stock de medicamento. El sistema se encarga de validar los datos ingresados del medicamento. En caso de existir algún error de escritura, no continúa el proceso, caso contrario, verifica que el medicamento este

registrado, si es así, en la base de datos almacena el stock de dicho medicamento, caso contrario, primero almacena los datos del medicamento y luego del stock del medicamento (Figura 2.10).

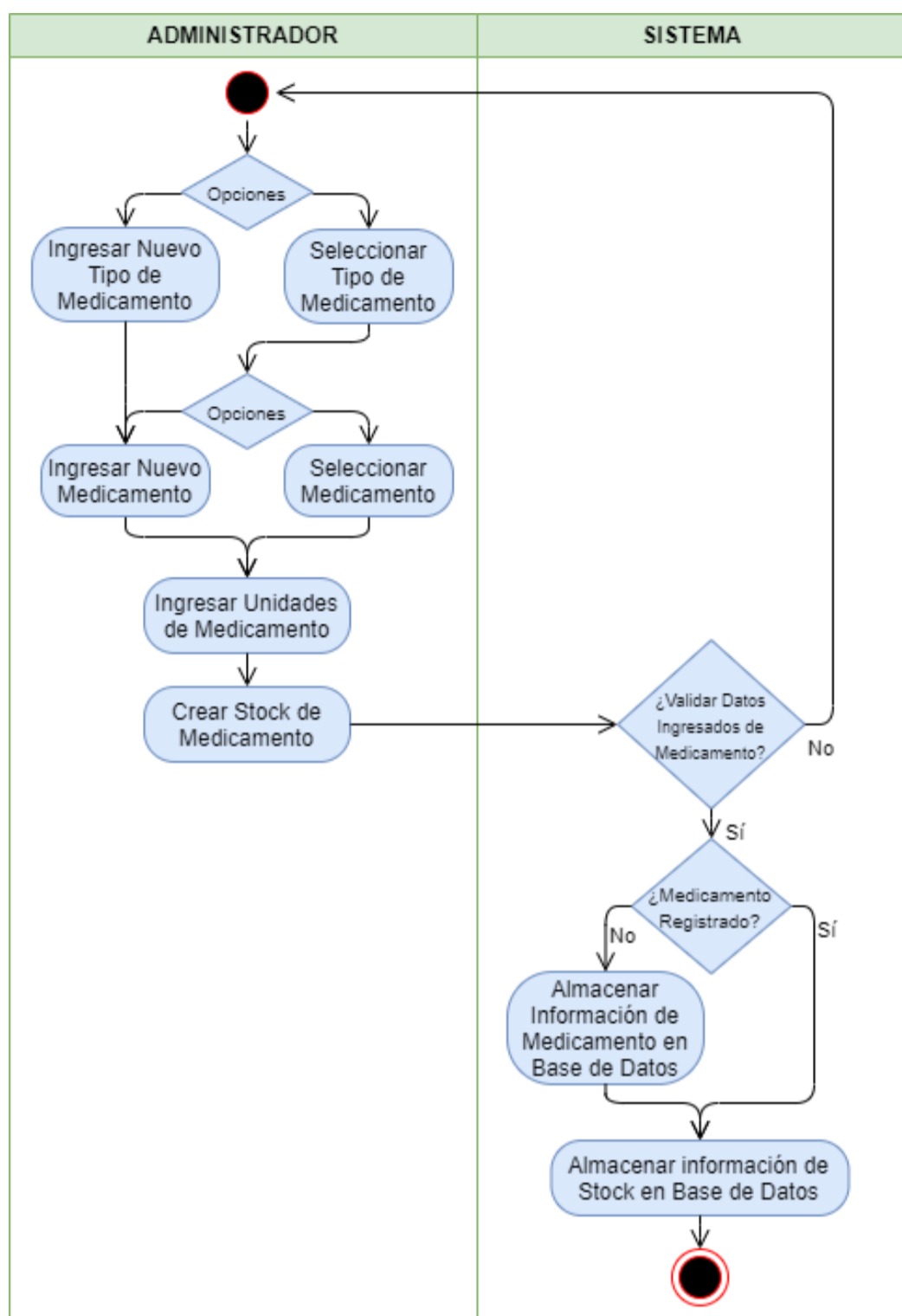


Figura 2.10. Diagrama de Actividades: Nuevo Stock de Medicamento

- **Lista de Medicamentos en Stock:**

El sistema inicia el proceso realizando la búsqueda de los medicamentos registrados en stock por el administrador y, en caso de no existir registros, el administrador visualiza un mensaje y finaliza el proceso, caso contrario, visualiza una lista de medicamentos en stock registrados y tiene dos opciones a escoger: finalizar todo el proceso o en caso de detectar algún error en la cantidad de unidades registradas, seleccionar uno de los stocks de la lista para su eliminación. Para ello, el sistema verifica si se ha registrado la entrega de parte del stock de medicamento, si existe registro, el administrador visualiza un mensaje, caso contrario, el sistema elimina el stock de la base de datos y el administrador visualiza un mensaje. Finalmente, el administrador visualiza la lista de stocks actualizada y puede repetir el proceso para otro stock de la lista o dar por finalizado todo el proceso (Figura 2.11).

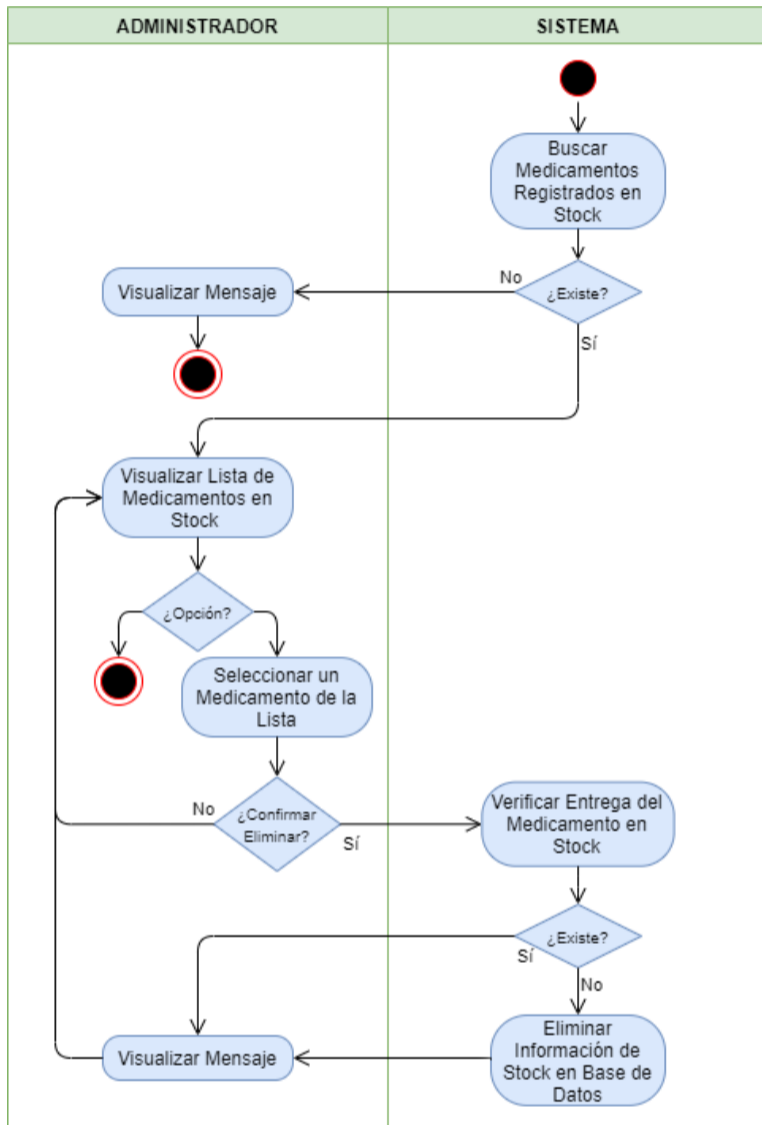


Figura 2.11. Diagrama de Actividades: Lista de Medicamentos en Stock

- **Historial Farmacéutico de Medicamento:**

Para iniciar el proceso, el administrador debe seleccionar un tipo de medicamento, luego seleccionar un medicamento, escoger el tipo de historial farmacéutico y generar dicho historial. El sistema valida dichos datos ingresados, en caso de existir algún error de escritura, no continúa el proceso, caso contrario, busca la información (entrada y salida) referente al medicamento. Si no existe la información, el administrador visualiza un mensaje, caso contrario, visualiza la información del medicamento. Finalmente, el administrador crea el documento Pdf con el historial del medicamento y el sistema se encarga de descargar dicho Pdf (Figura 2.12).

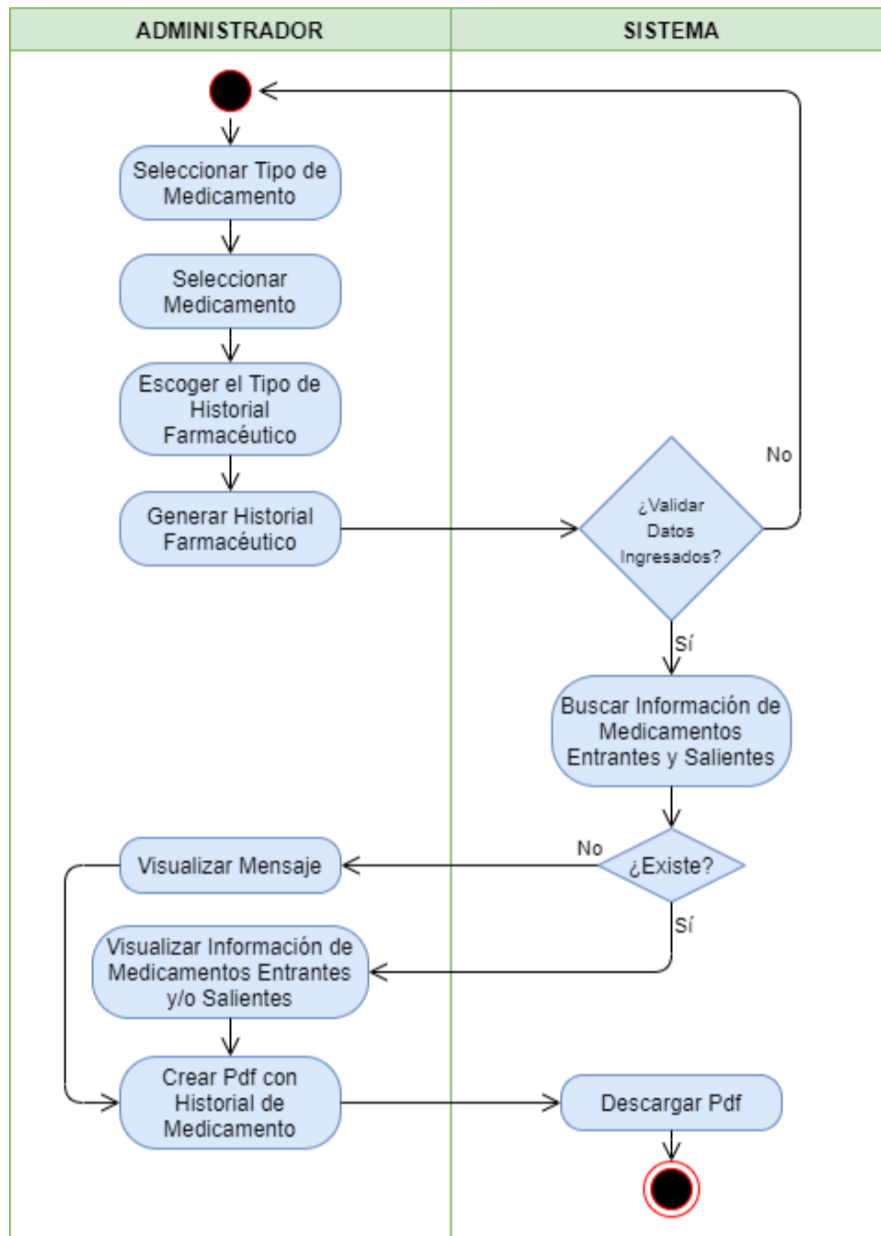


Figura 2.12. Diagrama de Actividades: Historial Farmacéutico de Medicamento

- **Nueva Atención Médica:**

Para iniciar el proceso, el médico debe ingresar la cédula de un paciente y buscar su registro. El sistema valida la cédula ingresada, en caso de existir algún error de escritura, no continúa con el proceso, caso contrario, obtiene los datos del paciente. En caso de existir el registro, el médico añade la receta, caso contrario, debe ingresar los datos personales del paciente: apellidos, nombres y sexo, y el sistema se encarga de validar dichos datos ingresados. En caso de existir algún error de escritura en los datos, no continúa el proceso, caso contrario, el médico añade la receta. Para la receta, debe seleccionar un medicamento y el sistema se encarga de validar los datos del medicamento, en caso de existir algún error de escritura o falta de stock del medicamento, no continúa el proceso, caso contrario, el médico visualiza el medicamento en la receta y tiene la opción de repetir este proceso para añadir más medicamentos a la receta. Al crear la atención médica, el sistema en la base de datos almacena los datos del paciente en caso de no estar registrado y, a continuación, almacena los datos de la atención médica y los medicamentos de la receta (Figura 2.13).

- **Atenciones Médicas Registradas:**

El sistema inicia el proceso realizando la búsqueda de las atenciones médicas registradas por el médico, en caso de no existir registros, el médico visualiza un mensaje y finaliza el proceso, caso contrario, el médico visualiza una lista de atenciones médicas y al seleccionar una de estas, el sistema busca los detalles y los visualiza, estos son: paciente y medicamentos de la receta. El médico puede finalizar el proceso o seleccionar un medicamento de la receta para actualizarlo o eliminarlo. Para actualizar, reemplaza el medicamento de la receta por otro medicamento y el sistema se encarga de validar los datos del medicamento, en caso de existir algún error de escritura en los datos o falta de stock de dicho medicamento, el médico visualiza un mensaje, caso contrario, el sistema actualiza el medicamento de la receta en la base de datos, el médico visualiza un mensaje y la lista de medicamentos actualizada. Para eliminar, el sistema verifica la fecha de entrega del mismo, si existe el registro, el médico visualiza un mensaje, caso contrario, el sistema elimina el medicamento de la receta en la base de datos y el médico visualiza un mensaje y la lista de medicamentos actualizados (Figura 2.14).

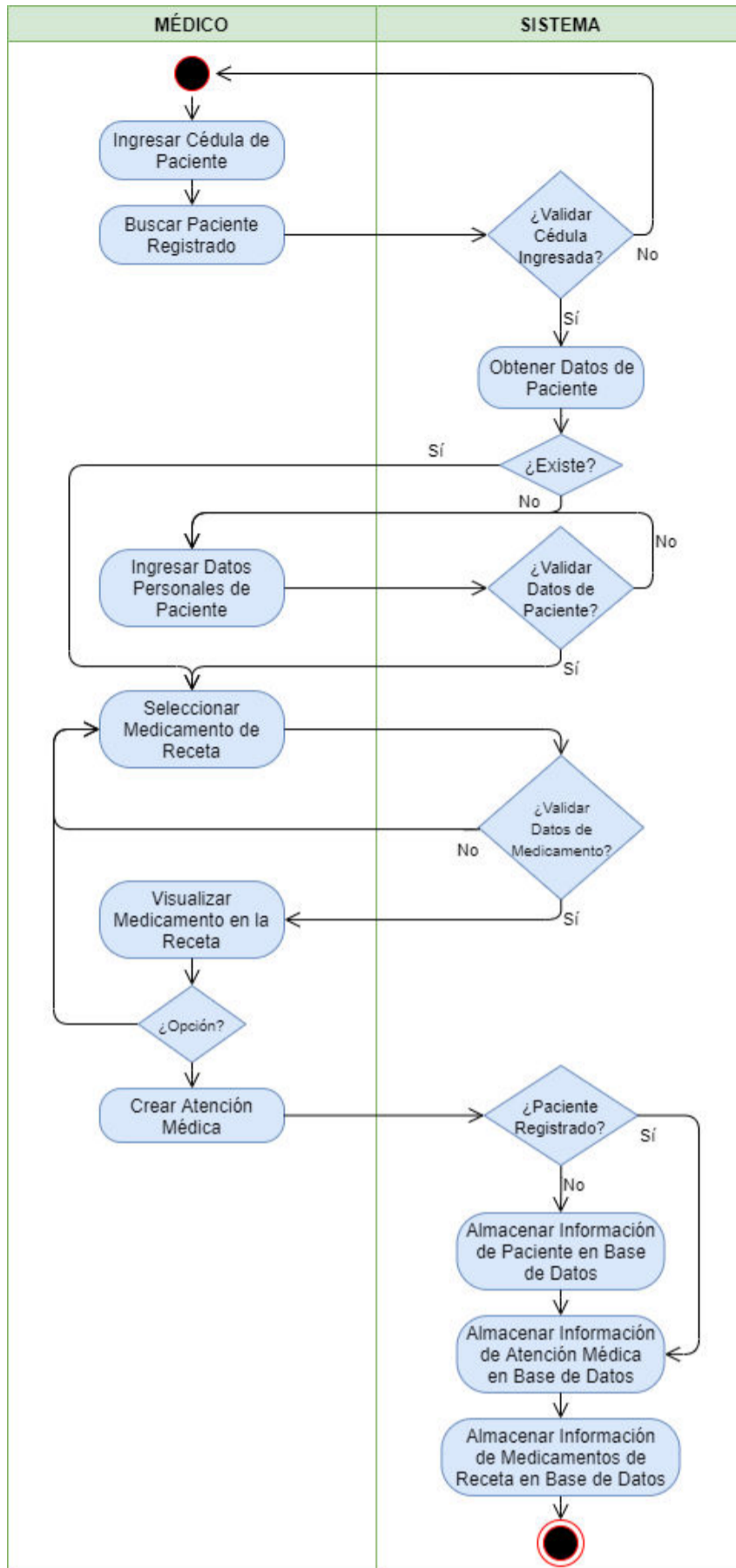


Figura 2.13. Diagrama de Actividades: Nueva Atención Médica

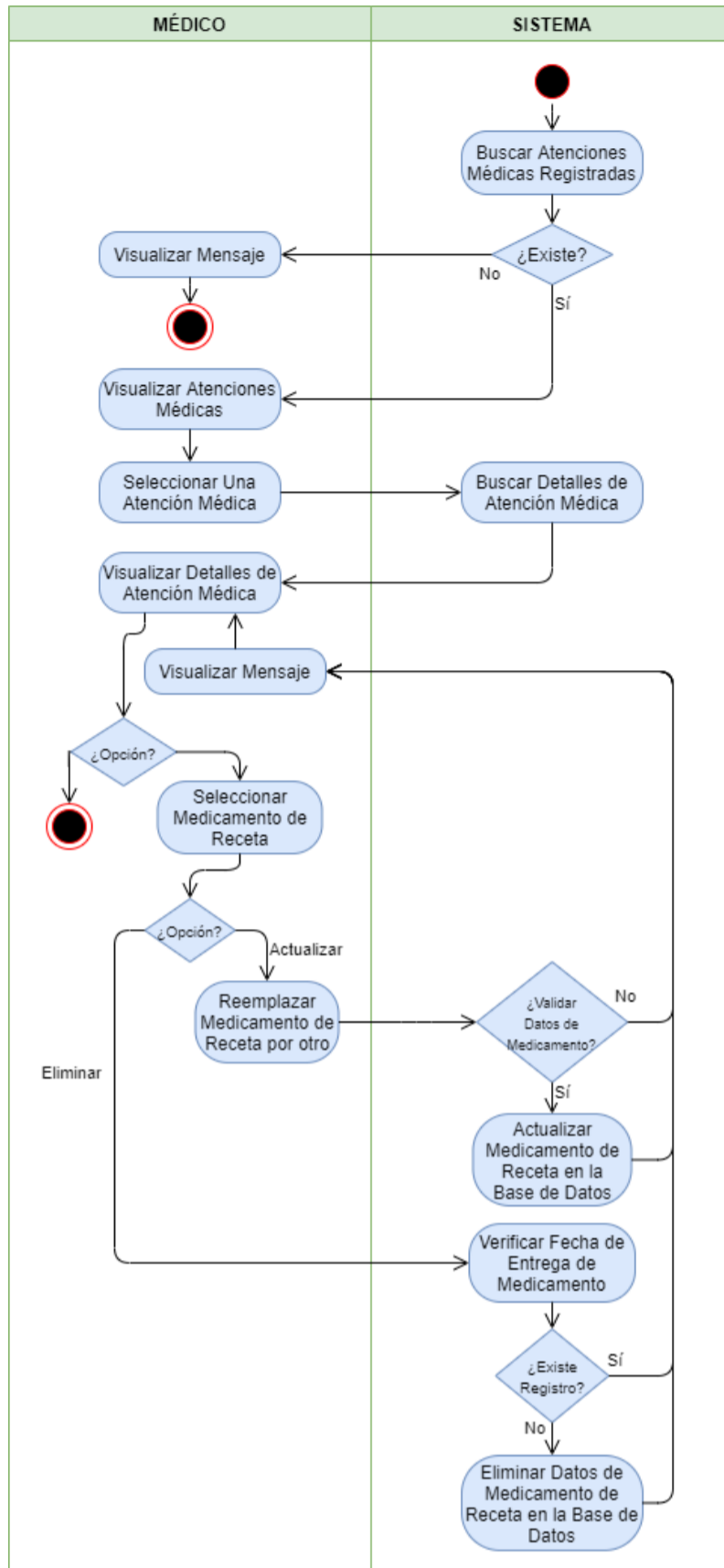


Figura 2.14. Diagrama de Actividades: Atenciones Médicas Registradas

- **Atenciones Médicas Por Entregar Medicamentos:**

Para iniciar el proceso, el dependiente de farmacia debe ingresar la cédula de un paciente y buscar su registro. El sistema valida la cédula ingresada, en caso de existir algún error de escritura, no continúa con el proceso, caso contrario, obtiene los datos del paciente y las atenciones médicas que registra dicho paciente, en caso de no existir los datos, el dependiente visualiza un mensaje y finaliza el proceso, caso contrario, visualiza los datos del paciente y las atenciones médicas que requieren entrega de medicamentos. Al seleccionar una atención médica el sistema busca sus detalles: médico y medicamentos de receta, y el dependiente los visualiza.

A continuación, el dependiente selecciona un medicamento de la receta para registrar su entrega y el sistema verifica la existencia de stock del medicamento. En caso de no existir, el dependiente visualiza un mensaje de no existencia de stock, caso contrario, el sistema almacena la fecha de entrega del medicamento en la base de datos y el dependiente visualiza un mensaje de registro exitoso. Debe repetir este proceso para cada medicamento de la receta. Al finalizar, el dependiente actualiza la atención médica y el sistema cambia su estado en la base de datos (Figura 2.15).

- **Atenciones Médicas Finalizadas:**

El sistema inicia el proceso realizando la búsqueda de las atenciones médicas finalizadas, es decir, en las que se ha registrado la entrega de todos los medicamentos de la receta. En caso de no existir registros, el dependiente de farmacia visualiza un mensaje y finaliza el proceso, caso contrario, visualiza una lista de atenciones médicas y al seleccionar una de ellas, el sistema busca los detalles y el dependiente los visualiza, estos son: médico, paciente y medicamentos de la receta (Figura 2.16).

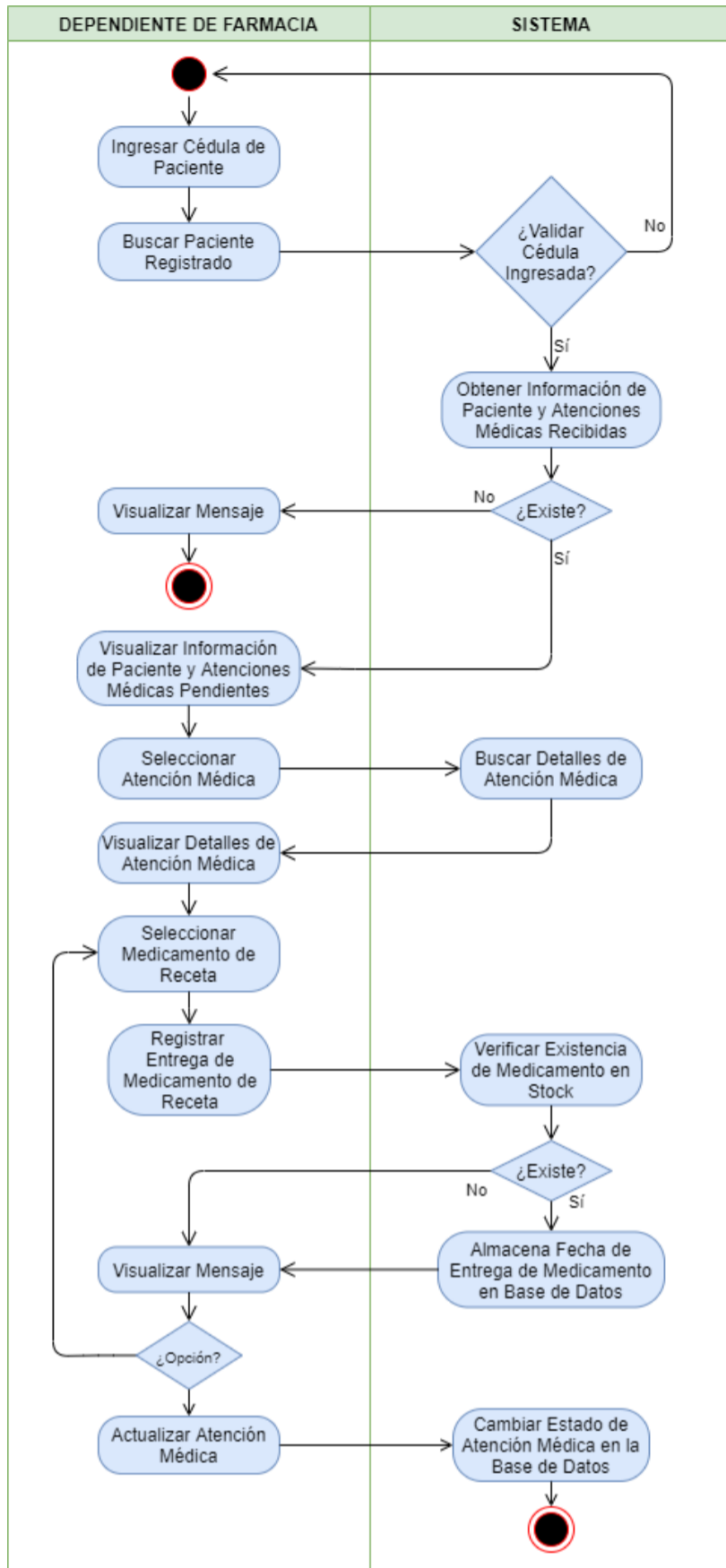


Figura 2.15. Diagrama de Actividades: Atenciones Médicas Por Entregar Medicamentos

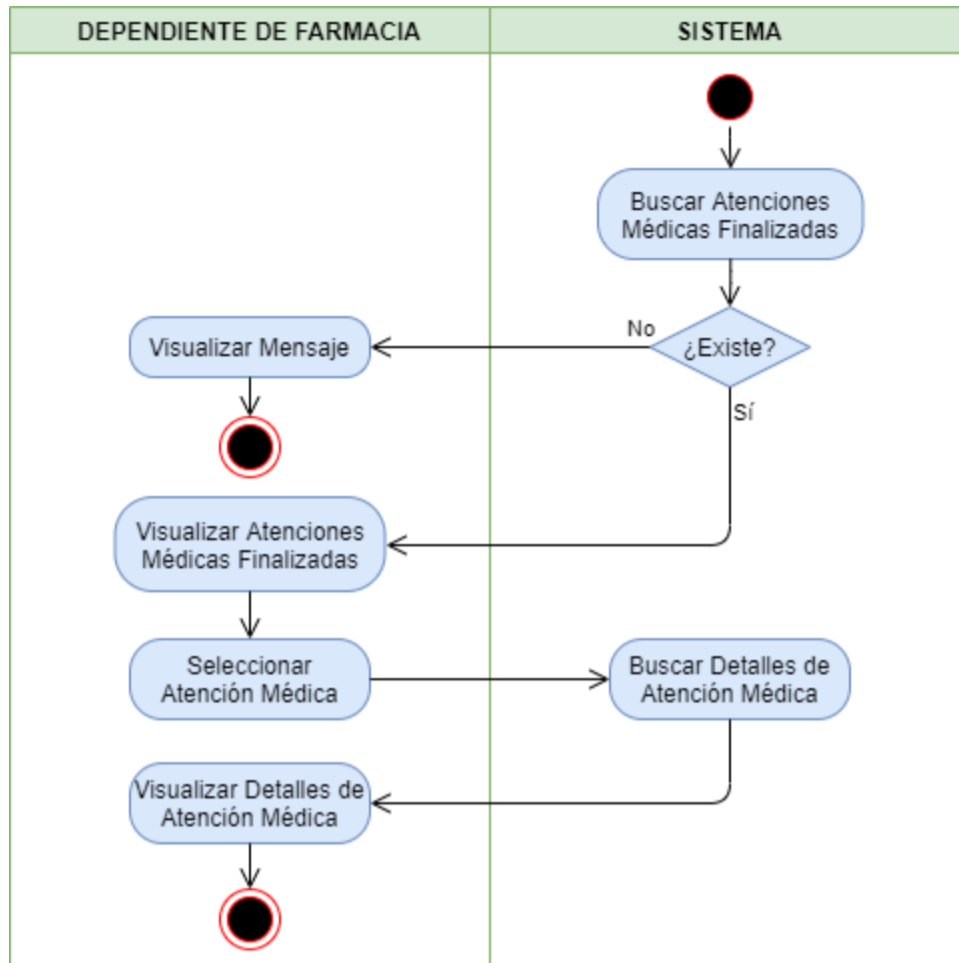


Figura 2.16. Diagrama de Actividades: Atenciones Médicas Finalizadas

2.1.6.3 DIAGRAMA DE CLASES

El diagrama de clases (Figura 2.17) representa la estructura y el comportamiento de cada uno de los objetos del sistema y sus relaciones con los demás objetos. Las clases contienen atributos y métodos [47].

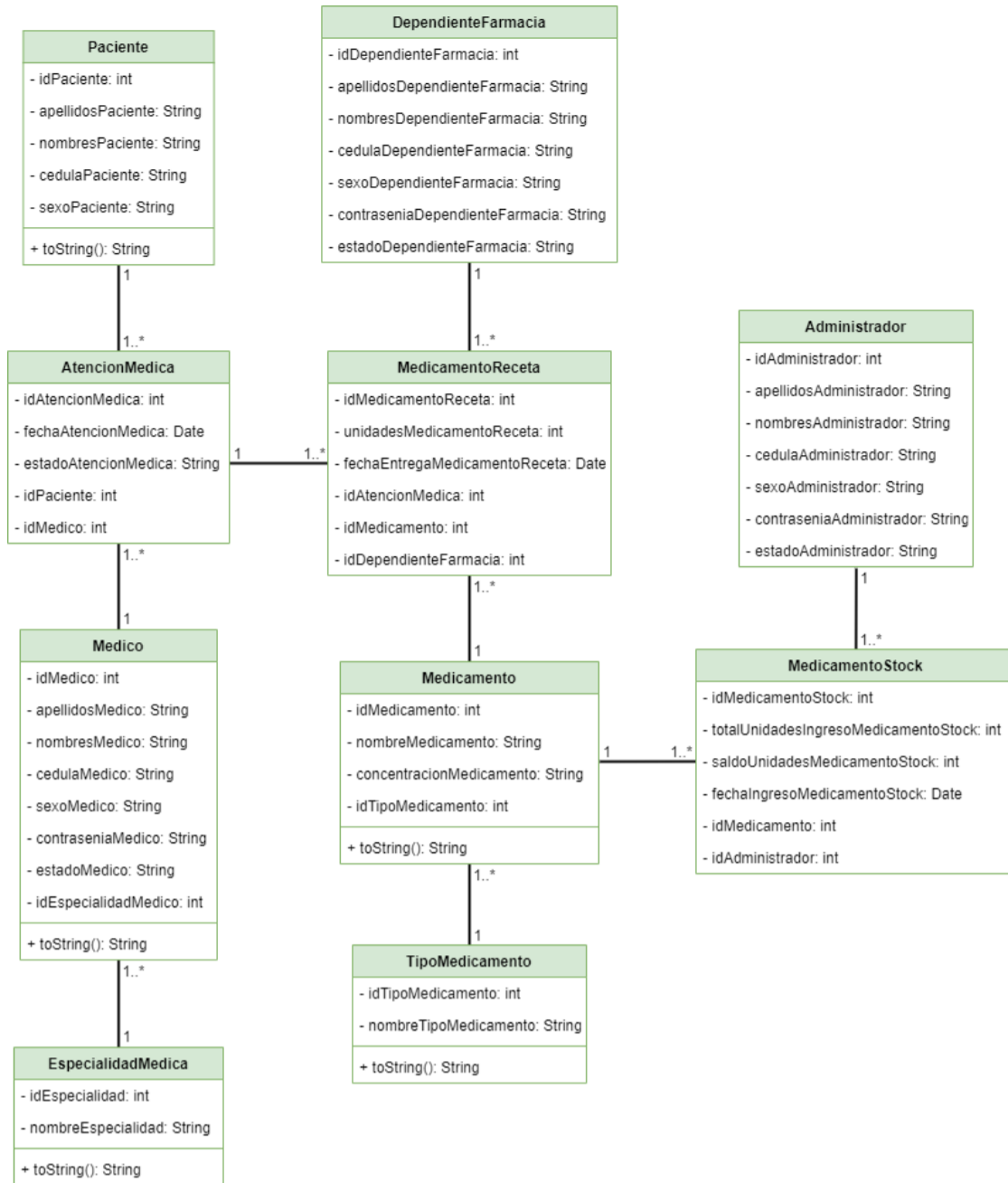


Figura 2.17. Diagrama de Clases

2.1.7 DISEÑO DE CAPA DE PRESENTACIÓN

La capa de presentación es la interfaz gráfica de la aplicación móvil, con la cual el usuario final interactúa, por lo que debe ser de fácil entendimiento y diseño amigable.

Para el presente proyecto, la aplicación móvil está compuesta por múltiples sketches gráficos, las cuales están diseñadas de acuerdo a los requerimientos para permitir la correcta funcionalidad del prototipo. A continuación, se presenta el diseño de cada sketch:

- **Módulo Registro de Usuario:**

Este módulo (Figura 2.18) permite al usuario registrar sus datos personales y consta de ocho campos: los tres primeros campos son para ingresar: apellidos, nombres y cédula, el cuarto campo es para seleccionar el sexo, el quinto campo es para seleccionar el tipo de usuario: Administrador, Médico o Dependiente de Farmacia, el sexto campo es para seleccionar la especialidad si el tipo de usuario es Médico, el séptimo campo es para ingresar una contraseña y el octavo campo es para confirmar la contraseña. Finalmente, cuenta con un botón “Registrarse” para almacenar la información en la base de datos. Existe un sólo Administrador registrado activo.

The sketch shows a mobile application interface for user registration. The title is "Registro de Usuario". The form consists of the following elements from top to bottom: three text input fields labeled "Apellidos", "Nombres", and "Cédula"; a "Sexo" section with two radio buttons, "Femenino" and "Masculino"; a dropdown menu for "Tipo de Usuario"; another dropdown menu for "Especialidad"; two text input fields for "Contraseña" and "Confirmar Contraseña"; and a blue button labeled "Registrarse" at the bottom.

Figura 2.18. Sketch Registro de Usuario

- El usuario Administrador una vez registrado en la aplicación móvil, es responsable de comprobar los usuarios registrados de tipo Médico y Dependiente de Farmacia, y activar sólo a aquellos que han sido facultados para acceder a las funcionalidades de la aplicación móvil o, por el contrario, desactivarlos o eliminarlos. Para ello, la interfaz (Figura 2.19) dispone de un campo de texto para seleccionar el tipo de usuario: Médico o Dependiente de Farmacia. Además, cuenta con una lista donde se despliegan los usuarios registrados de acuerdo al tipo de usuario. La selección de un usuario permite realizar el proceso de activación, desactivación o eliminación de dicho usuario.

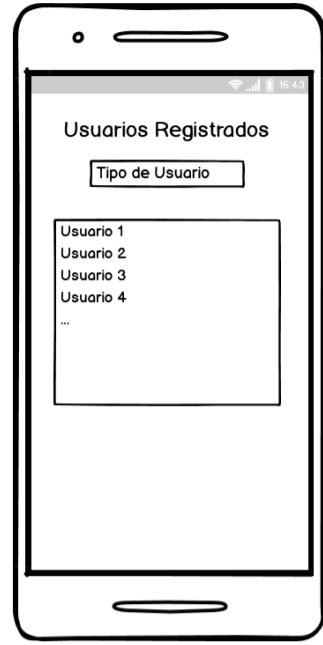


Figura 2.19. Sketch Usuarios Registrados

- **Módulo Inicio de Sesión de Usuario:**

Este módulo (Figura 2.20) permite al usuario registrado previamente, ingresar a las funcionalidades correspondientes, consta de tres campos: el primero para seleccionar el tipo de usuario y los dos siguientes para ingresar la cédula y contraseña. Finalmente, dispone de dos botones: el primero "Iniciar" para acceder al menú de usuario correspondiente y el segundo "Registrarse" para acceder al módulo de Registro de Usuario.



Figura 2.20. Sketch Inicio de Sesión de Usuario

- **Menú Administrador**


El usuario administrador dispone de una interfaz principal (Figura 2.21), que está conformada por cinco botones y cada uno permite acceder al módulo correspondiente.




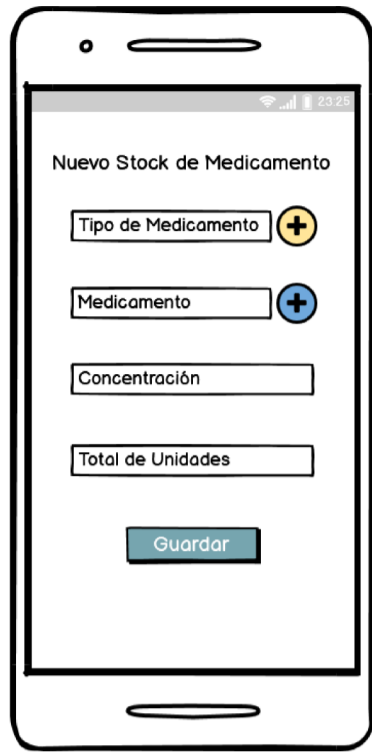
Figura 2.21. Sketch Menú Administrador

- **Módulo Nuevo Stock de Medicamento:**

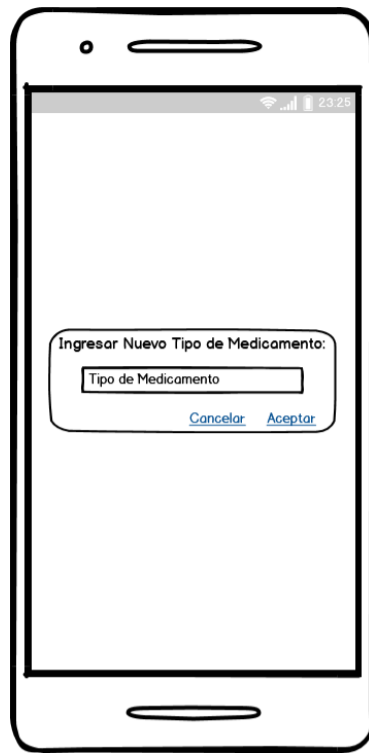
Este módulo (Figura 2.22) permite al usuario administrador registrar un nuevo stock de un medicamento específico (Parte a), consta de cuatro campos: tipo de medicamento, medicamento, concentración y cantidad de unidades del medicamento. Además, cuenta con un botón “Guardar” para registrar la información en la base de datos.

El botón circular  ubicado a la derecha del campo Tipo de Medicamento, despliega un cuadro de dialogo (Parte b), que está conformado por un campo para ingresar un nuevo tipo de medicamento. Dicha información se registra en la base de datos.

El botón circular  ubicado a la derecha del campo Medicamento, despliega un cuadro de dialogo (Parte c) para ingresar los datos de un nuevo medicamento, está conformado por dos campos, el primero para ingresar el nombre de un nuevo medicamento y en el segundo su concentración. Dicha información se registra en la base de datos.



(a)



(b)



(c)

Figura 2.22. Sketch Nuevo Stock de Medicamento

- **Módulo Lista de Medicamentos en Stock:**

Este módulo (Figura 2.23) muestra la lista de medicamentos en stock que han sido registrados por el usuario administrador, y al seleccionar un medicamento de la lista puede eliminarlo, pero sólo en caso de no haberse registrado ninguna entrega de dicho medicamento.



Figura 2.23. Sketch Lista de Medicamentos en Stock

- **Módulo Historial Farmacéutico de Medicamento:**

El módulo (Figura 2.24) permite al administrador generar el historial farmacéutico de un medicamento con tres opciones posibles: Total, Anual y Mensual. Este módulo está conformado por dos interfaces gráficas. La primera interfaz (Parte a) consta de cinco campos: el primero para seleccionar el tipo de medicamento, el segundo para seleccionar el medicamento, el tercero para seleccionar el tipo de historial del medicamento, donde se puede elegir entre tres opciones: Total, Anual o Mensual, el cuarto campo para ingresar el año, si el tipo de historial es Anual o Mensual y el quinto campo para seleccionar el mes, si el tipo de historial es Mensual. En la parte inferior cuenta con un botón "Generar Historial", que permite acceder a la segunda interfaz. La segunda interfaz (Parte b) permite visualizar el resultado del historial generado, para ello, cuenta con cinco campos para visualizar los datos del medicamento y el tipo de historial, dos tablas que se generan con los datos obtenidos de la base de datos correspondientes a los medicamentos entrantes y salientes, y dos campos que corresponden al saldo de unidades del medicamento entrante y saliente. Además, dispone en la parte inferior de un botón "Generar PDF" para crear

el documento en formato PDF con la información correspondiente al historial farmacéutico del medicamento.

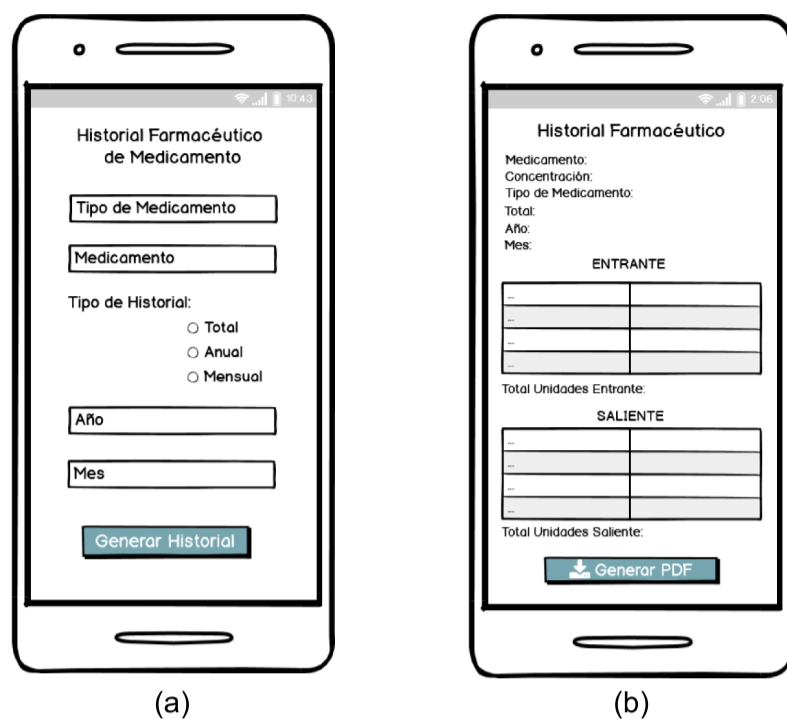


Figura 2.24. Sketch Historial Farmacéutico de Medicamento

- **Menú Médico:**

El usuario médico dispone de una interfaz principal (Figura 2.25), que está conformada por tres botones y cada uno permite acceder al módulo correspondiente.



Figura 2.25. Sketch Menú Médico

- **Módulo Nueva Atención Médica:**

Este módulo (Figura 2.26) permite al médico registrar una atención médica realizada a un paciente y está conformado por dos interfaces gráficas para dicho proceso.

La primera interfaz (Parte a) es para ingresar los datos de un paciente, cuenta con cuatro campos: los tres primeros son para la cédula, apellidos y nombres del paciente y en el cuarto se debe seleccionar el sexo. Además, dispone de un botón “Siguiente” que permite acceder a la segunda interfaz. La segunda interfaz (Parte b) es para agregar medicamentos a la receta, por lo que en la parte superior consta de tres campos: el primero para seleccionar el tipo de medicamento, el segundo para seleccionar el medicamento, en el tercer campo se debe ingresar la cantidad de unidades del medicamento. También cuenta con un botón “Agregar”, que permite agregar dicho medicamento a la receta y puede repetir el proceso para añadir más medicamentos a dicha receta. A continuación, dispone de una lista para visualizar los medicamentos agregados a la receta y en la parte inferior consta de un botón “Finalizar Receta” para registrar los datos del paciente y la atención médica.

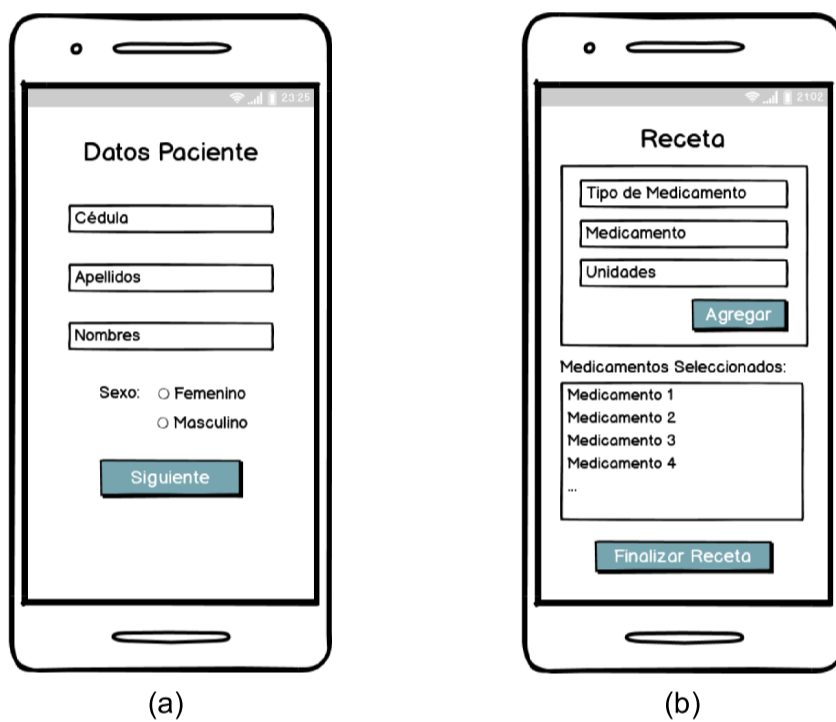


Figura 2.26. Sketch Nueva Atención Médica

- **Módulo Atenciones Médicas Registradas:**

Este módulo (Figura 2.27) permite visualizar todas las atenciones médicas que han sido registradas por el médico y está conformado por dos interfaces gráficas.

En la primera interfaz (Parte a) el médico visualiza la lista de atenciones médicas registradas y al seleccionar una atención médica se accede a la segunda interfaz para visualizar los detalles de dicha atención médica.

En la segunda interfaz (Parte b) se despliega la información sobre la atención médica, en la que se puede visualizar la fecha de la atención médica, los datos personales del paciente, el nombre del médico, especialidad y la lista de medicamentos de la receta. Al seleccionar un medicamento de la lista se puede eliminar o actualizar dicho medicamento por otro. Para el proceso de actualización se despliega una sección en la parte inferior que consta de tres campos, el primero para seleccionar el tipo de medicamento, el segundo para seleccionar el nuevo medicamento, en el tercer campo debe ingresar la cantidad de unidades del medicamento. Además, esta sección consta de dos botones “Actualizar” y “Cancelar”, el primero permite actualizar el medicamento de la receta y el segundo para cancelar dicha acción.

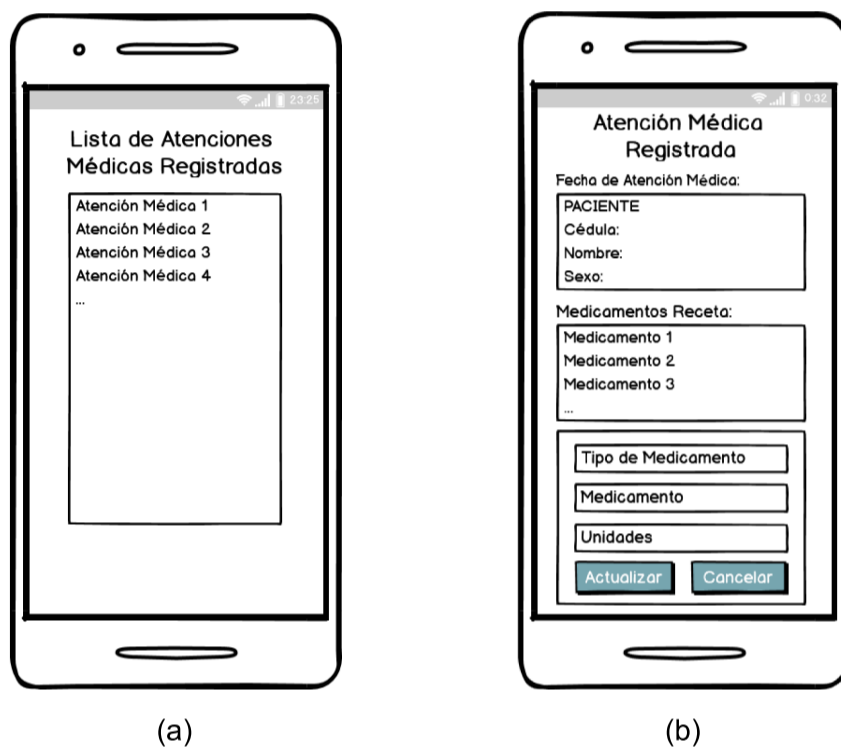


Figura 2.27. Sketch Atenciones Médicas Registradas

- **Menú Dependiente de Farmacia:**

El usuario dependiente de farmacia dispone de una interfaz principal (Figura 2.28), que está conformada por tres botones para acceder al módulo correspondiente.



Figura 2.28. Sketch Menú Dependiente de Farmacia

- **Módulo Atenciones Médicas Por Entregar Medicamentos:**

Este módulo (Figura 2.29) permite al dependiente de farmacia visualizar las atenciones médicas que no tienen registrado la entrega de medicamentos de la receta, para registrar dicha entrega y está conformado por dos interfaces gráficas.

La primera interfaz (Parte a) contiene un campo para ingresar la cédula del paciente, a la derecha dispone de un botón circular para realizar la búsqueda del paciente y las atenciones médicas recibidas por dicho paciente. Seguidamente, consta de un campo para visualizar los datos del paciente: apellidos y nombres. Finalmente, dispone de una lista donde se despliegan las atenciones médicas que tienen medicamentos por entregar. La selección de una atención médica permite acceder a la segunda interfaz.

En la segunda interfaz (Parte b) se despliega la información sobre la atención médica, en la que se puede visualizar los datos personales del paciente, el nombre del médico, especialidad y la lista de medicamentos de la receta. Al seleccionar un medicamento se puede registrar la entrega del mismo, en caso de no existencia en stock, no se registra la entrega de dicho medicamento. Además, consta de un botón "Finalizar Entrega" en la parte inferior para finalizar el proceso.

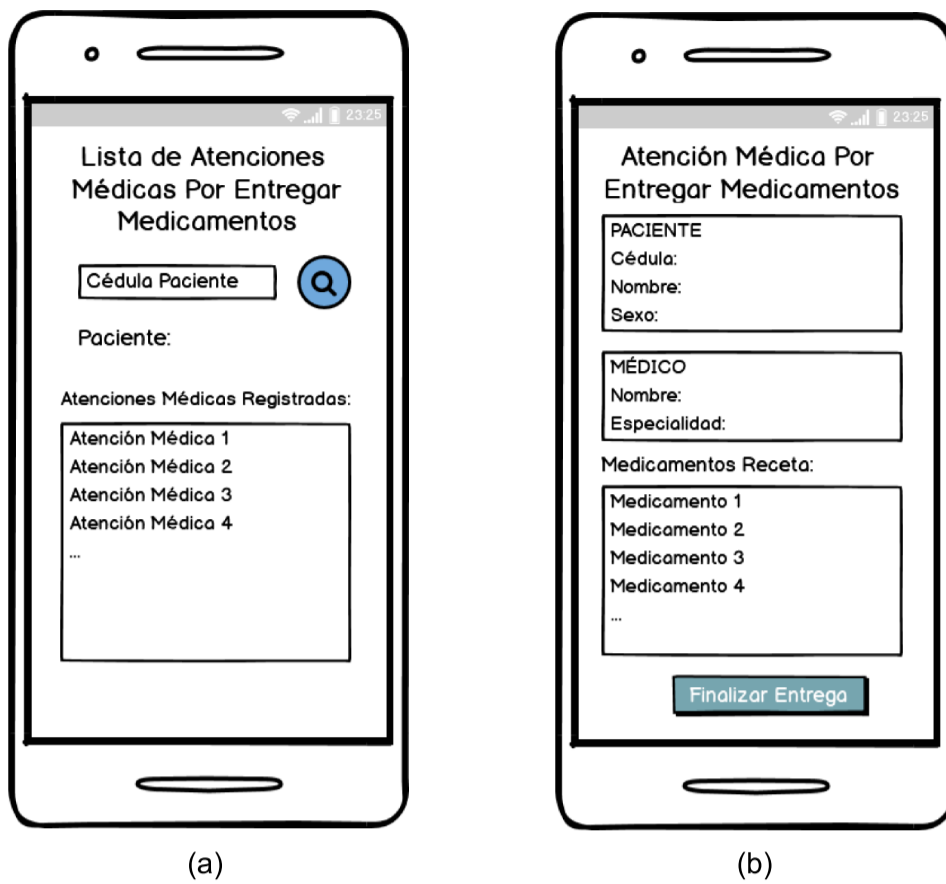


Figura 2.29. Sketch Atenciones Médicas Por Entregar Medicamentos

- **Módulo Atenciones Médicas Finalizadas:**

Este módulo (Figura 2.30) permite al dependiente de farmacia visualizar las atenciones médicas que tienen registrado la entrega de todos los medicamentos de la receta y está conformado por dos interfaces gráficas.

La primera interfaz (Parte a) permite visualizar la lista de todas las atenciones médicas en las que se ha registrado la entrega de todos los medicamentos de receta. Al seleccionar una de estas atenciones médicas se accede a la segunda interfaz.

En la segunda interfaz (Parte b) se despliega la información sobre la atención médica, en la que se puede visualizar los datos personales del paciente, el nombre del médico, especialidad y la lista de medicamentos de la receta que registran una fecha de entrega.

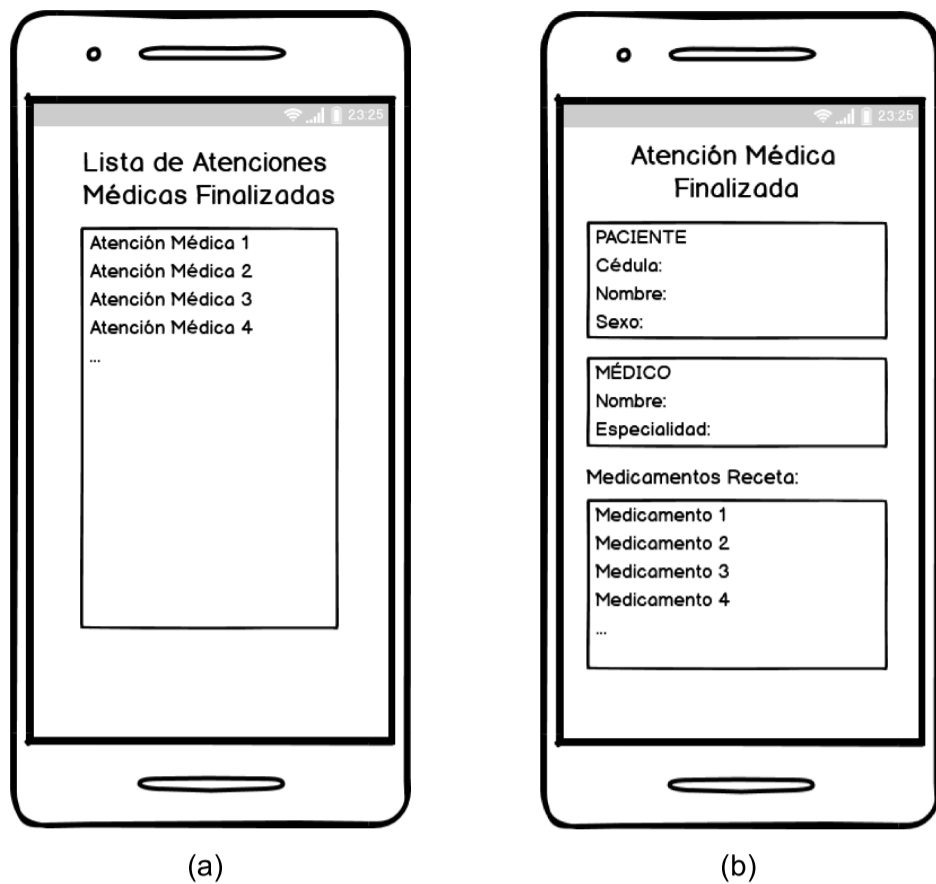


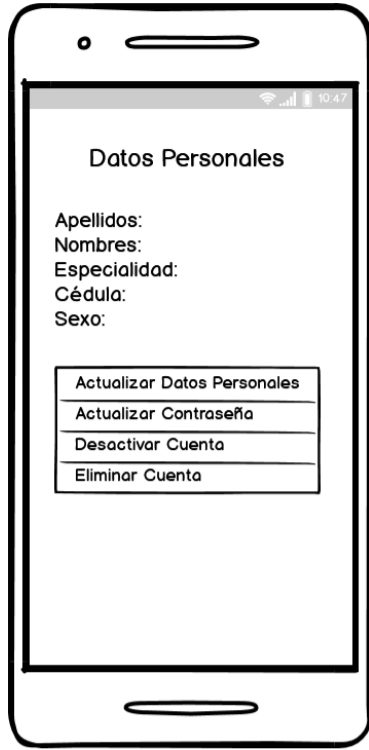
Figura 2.30. Sketch Atenciones Médicas Finalizadas

- **Módulo Datos Personales de Usuario:**

Este módulo (Figura 2.31) está conformado por tres interfaces gráficas. La primera interfaz (Parte a) permite visualizar los datos personales de usuario: apellidos, nombres, especialidad, cédula y sexo, y un menú de opciones para realizar los procesos de: actualizar datos personales, actualizar contraseña, desactivar cuenta y eliminar cuenta.

La segunda interfaz (Parte b) consta de dos campos para ingresar: apellidos y nombres, un campo para seleccionar el sexo y un botón “Actualizar” para actualizar los datos del usuario.

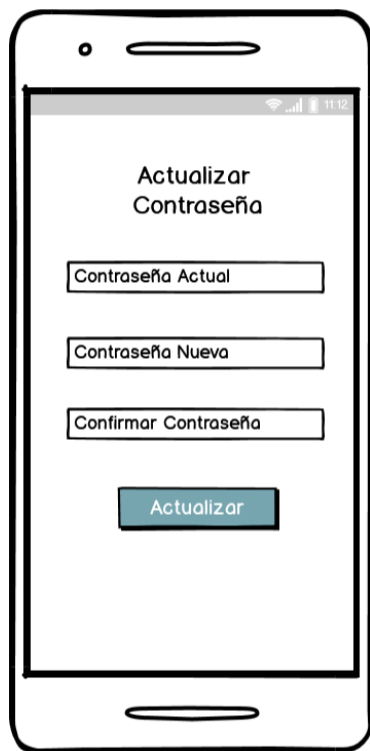
La tercera interfaz (Parte c) consta de tres campos para ingresar: contraseña actual, contraseña nueva y confirmar contraseña, y un botón “Actualizar” para actualizar la contraseña del usuario.



(a)



(b)



(c)

Figura 2.31. Sketch Datos Personales de Usuario

2.2 IMPLEMENTACIÓN

En esta sección primero se realiza la actualización del tablero Kanban y a continuación, para la implementación del prototipo se realiza la configuración de AWS para acceder a los servicios en la nube y empleando las herramientas de software necesarias se realiza la codificación de las tres capas: datos, negocio y presentación.

2.2.1 ACTUALIZACIÓN DEL TABLERO KANBAN

En la primera actualización de actividades del tablero Kanban (Tabla 2.24), las tareas que se encontraban en proceso de desarrollo han sido culminadas, por lo que ahora se encuentran en la columna de tareas realizadas y las tareas que estaban pendientes, ahora se encuentran en proceso de desarrollo, a excepción de las pruebas de funcionamiento del prototipo y las encuestas de validación del funcionamiento a usuarios que todavía se encuentran pendientes de desarrollo.

Tabla 2.24. Primera Actualización de Tablero Kanban

Tareas Pendientes	Tareas en Proceso	Tareas Realizadas
Pruebas de funcionamiento del prototipo	Creación de una instancia de EC2 con Windows Server	Definición de conceptos teóricos
Encuestas de validación de funcionamiento a usuarios	Creación de una instancia de RDS con MySql	Entrevistas a administrador, personal médico y dependiente de farmacia del centro de salud
	Instalación de Node.js y Visual Studio Code en Windows Server	Identificación de Requerimientos Funcionales
	Instalación de Android Studio	Identificación de Requerimientos No Funcionales
	Codificación de Capa de Datos	Definición de módulos de la aplicación móvil
	Codificación de Capa de Negocio	Diseño de Capa de Datos
	Codificación de Capa de Presentación	Diseño de Capa de Negocio
		Diseño de Capa de Presentación

2.2.2 CONFIGURACIÓN DE AWS

Para acceder a los servicios en la nube requeridos para el proyecto, en la plataforma de Amazon Web Services se crearon dos tipos de instancias: EC2 y RDS, las cuales permiten el alojamiento del servicio web y la base de datos en la nube.

Instancia EC2:

- Se seleccionó una instancia de Windows Server (Figura 2.32) para alojar el servicio web.

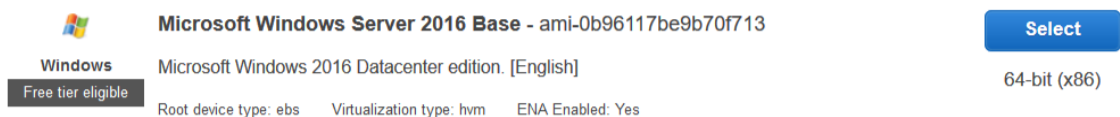


Figura 2.32. Instancia de EC2 Seleccionada

- El volumen de almacenamiento de la instancia es de 30GB (Figura 2.33), establecido por defecto en un servicio Free tier.



Figura 2.33. Volumen de Instancia EC2

- Una vez creada la instancia (Figura 2.34), esta se encuentra inicializada y consta de una dirección pública para poder acceder al servidor.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP
i-029deb4bd...	i-029deb4bd...	t2.micro	sa-east-1a	running	2/2 check...	None	ec2-18-230-116-131.sa-eas...	18.230.116.131

Figura 2.34. Instancia de EC2 Creada

Instancia RDS:

- Se seleccionó una instancia de MySQL con su respectiva versión (Figura 2.35) para alojar la base de datos.

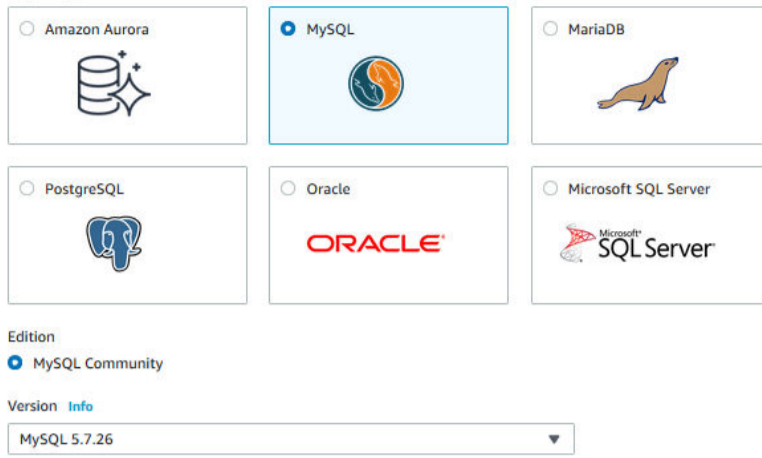


Figura 2.35. Instancia de RDS Seleccionada

- Se seleccionó el tipo de instancia Free Tier (Figura 2.36), para tener acceso libre de pago por un año.



Figura 2.36. Modelo Free Tier de Instancia RDS

- Se asignó el nombre de la instancia, así como las credenciales: usuario y contraseña (Figura 2.37).

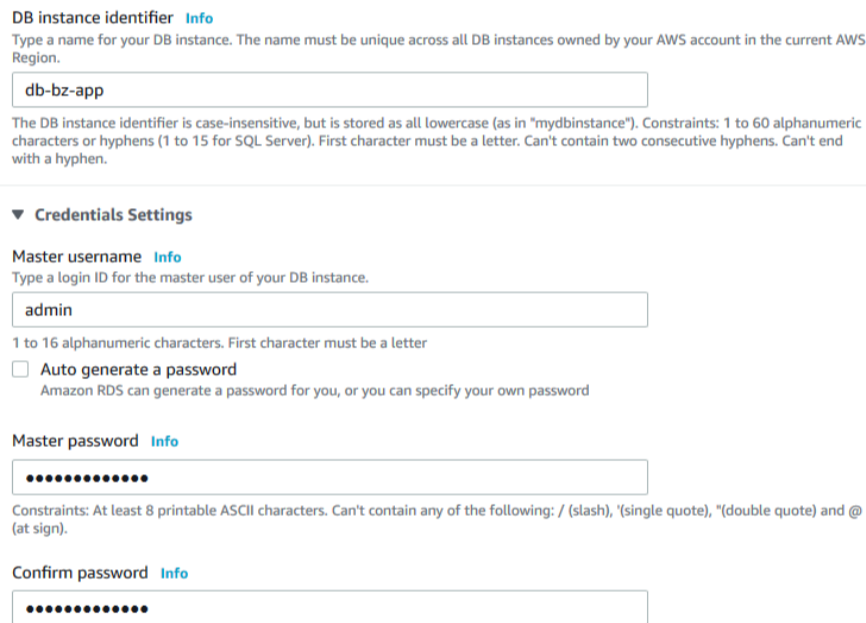


Figura 2.37. Identificación de Instancia RDS

- El volumen de almacenamiento de la instancia es de 20GB (Figura 2.38), establecido por defecto en un servicio Free tier y es autoescalable.

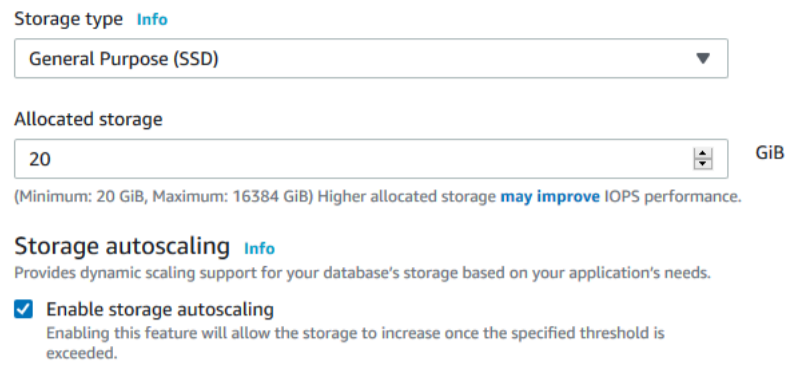


Figura 2.38. Volumen de Instancia RDS

- Una vez creada la instancia (Figura 2.39), esta se encuentra disponible para la creación de la base de datos y almacenar la información en la misma.

DB identifier	Role	Engine	Class	Status
db-bz-app	Instance	MySQL Community	db.t2.micro	Available

Figura 2.39. Instancia de RDS Creada

2.2.3 CODIFICACIÓN

2.2.3.1 CODIFICACIÓN DE CAPA DE DATOS

El proceso de codificación de la capa de datos se realizó de forma local antes de migrar al Cloud Computing de AWS. En MySQL Workbench se estableció una conexión local para la creación de la base de datos (Figura 2.40).

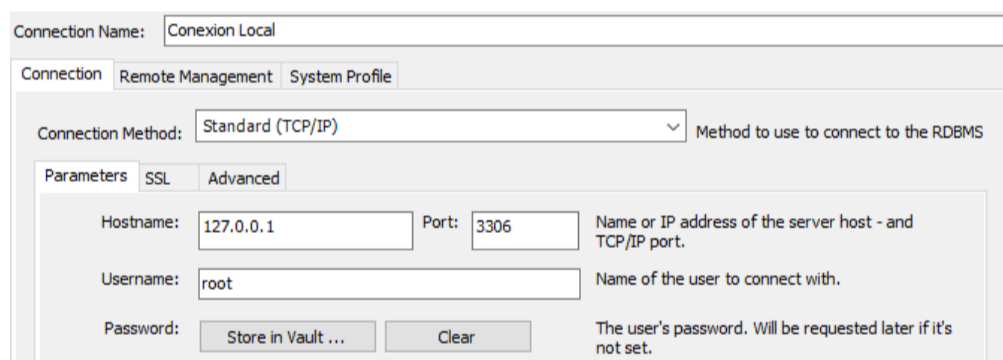


Figura 2.40. Conexión Local de Base de Datos

Para la codificación de esta capa se empleó un script para la creación de la base de datos y sus tablas relacionadas. En el Código 2.1, en la línea 1 se crea la base de datos db_app_ciec y en la línea 2 se designa la base de datos a utilizarse.

```
1 create database db_app_ciec;
2 use db_app_ciec;
```

Código 2.1. Creación de Base de Datos

Una vez creada la base de datos, se crean las tablas que contiene la base de datos. En el Código 2.2 se presenta la creación de la tabla tbl_Tipo_Medicamento, en las líneas 2 y 3 se especifican los campos que contiene la tabla y en la línea 4 se designa al campo que se identifica como la llave primaria.

```
1 create table tbl_Tipo_Medicamento(
2     id_Tipo_Medicamento int not null auto_increment,
3     nombre_Tipo_Medicamento varchar(30) not null,
4     primary key id_Tipo_Medicamento (id_Tipo_Medicamento)
5 );
```

Código 2.2. Creación de Tabla tbl_Tipo_Medicamento

En el Código 2.3 se presenta la creación de la tabla tbl_Medicamento, entre las líneas 2 y 5 se especifican los campos que contiene la tabla, en la línea 6 se designa al campo que se identifica como la llave primaria, en la línea 7 se designa al campo que se identifica como la llave foránea estableciendo la referencia a la tabla tbl_Tipo_Medicamento (Código 2.2) con la cual está relacionada, demostrándose que la base de datos es relacional. En el ANEXO B se encuentra la codificación restante de la base de datos.

```
1 create table tbl_Medicamento(
2     id_Medicamento int not null auto_increment,
3     nombre_Medicamento varchar(50) not null,
4     concentracion_Medicamento varchar(50) not null,
5     id_Tipo_Medicamento int not null,
6     primary key id_Medicamento (id_Medicamento),
7     constraint fk_id_Tipo_Medicamento_tbl_Tipo_Medicamento foreign key
      (id_Tipo_Medicamento) references
      tbl_Tipo_Medicamento(id_Tipo_Medicamento) on DELETE CASCADE on
      UPDATE CASCADE
8 );
```

Código 2.3. Creación de Tabla tbl_Medicamento

2.2.3.2 CODIFICACIÓN DE CAPA DE NEGOCIO

Para la implementación de esta capa se realiza la codificación del servicio web y en Android Studio la codificación de las clases, el cliente Retrofit y la declaración de los métodos con las operaciones HTTP.

Servicio Web:

La codificación del servicio web se realizó de forma local antes de migrar al Cloud Computing de AWS y se empleó Node.js y el editor de código Visual Studio Code. En el editor de código se instaló el paquete package.json (Figura 2.41), el cual contiene la configuración e incluye los paquetes de los que depende el proyecto, versiones, dependencias, autor, licencia, etc.

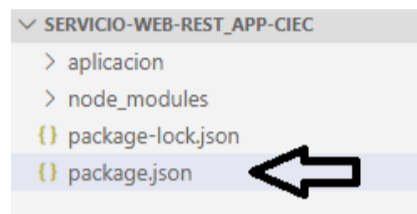


Figura 2.41. Archivo package.json

En el Código 2.4 se presenta el contenido del archivo package.json, el cual consta del nombre del proyecto, versión, descripción, autor y en la sección de dependencias se encuentran los tres módulos instalados: express, mysql y morgan.

```
1 {
2   "name": "servicio-web-rest_app-ciec",
3   "version": "1.0.0",
4   "description": "Servicio Web Rest - CIEC",
5   "main": "app.js",
6   "scripts": {
7     "test": "echo \"Error: no test specified\" && exit 1"
8   },
9   "author": "Andrés Marcelo Parra Acosta",
10  "license": "ISC",
11  "dependencies": {
12    "express": "^4.17.1",
13    "morgan": "^1.9.1",
14    "mysql": "^2.17.1"
15  }
16 }
```

Código 2.4. Configuración del package.json

El script principal (Código 2.5) se encarga de ejecutar el servidor y está codificado de la siguiente manera: en las líneas 1 y 2 se importa el módulo Express utilizando la función require() y se emplea la función para crear la aplicación Express, en las líneas 4 y 5 se importa el módulo Morgan y se establece el uso del registro de solicitudes y errores en la consola, en la línea 6 se especifica el puerto de escucha del servidor, en la línea 7 se emplea el método incorporado en express para reconocer el objeto de solicitud entrante como un objeto JSON, en la línea 9 se importa el script donde se encuentran definidas todas las rutas del servicio web y entre las líneas 11 y 13 se abre el puerto de escucha del servidor para las peticiones realizadas desde la aplicación móvil.

```
1  const express = require('express');
2  const app = express();
3
4  const morgan = require('morgan');
5  app.use(morgan('dev'));
6  app.set('puerto', 8080 );
7  app.use(express.json());
8
9  require('../rutas/rutas_acceso')(app);
10
11 app.listen(app.get('puerto'), () => {
12   console.log('Servidor en el puerto:', app.get('puerto') );
13 });
```

Código 2.5. Configuración del Servidor

El script (Código 2.6) contiene las rutas del servicio REST y emplea los métodos GET, POST, PUT Y DELETE de HTTP. Está codificado de la siguiente manera: en la línea 1 se importa el script donde se encuentran definidas todas las consultas SQL de la base de datos, en la línea 3 se define la función que se exporta para ser utilizada por el script principal, en la línea 4 se define una ruta empleando uno de los métodos de HTTP, en este caso GET, en la línea 5 se emplea una función definida para la conexión con la base de datos, en la línea 6 se emplea un condicional if y en caso de obtener los datos, en la línea 7 el servidor los envía en formato Json, caso contrario, en las líneas 9 y 10 envía un código de error y un mensaje en formato Json. Las demás rutas del servicio se encuentran definidas en el ANEXO C.

```

1  const miConexion = require('../conexion_base_datos/conexion_sql');
2
3  module.exports = function(app){
4    app.get('/obtenerEspecialidadesMedicas', (req, res) =>{
5      miConexion.obtenerEspecialidadesMedicas((error, datos) =>{
6        if(datos){
7          res.json(datos);
8        } else {
9          res.status(404).json({
10           mensaje: 'Error al obtener Especialidades Médicas'
11         });
12       }
13     });
14   });
15 }

```

Código 2.6. Ruta de Servicio obtenerEspecialidadesMedicas

El script (Código 2.7) contiene la conexión a la base de datos y las consultas SQL. Está codificado de la siguiente manera: en la línea 1 se importa el módulo Mysql utilizando la función require(), entre las líneas 3 y 8 se establece una conexión a la base de datos, para lo cual se define el host donde se encuentra la base de datos, el nombre de usuario, la contraseña y el nombre de la base de datos. A continuación, en la línea 10 se define un objeto, en la línea 11 se define una propiedad de dicho objeto que contiene la función callback como parámetro, en la línea 12 se establece un condicional if para determinar la conexión con la base de datos, en la línea 13 se define la sentencia SQL que se ejecuta en la base de datos si se estableció conexión con la misma, en la línea 14 se establece un condicional if para determinar un error al obtener la respuesta de la base de datos, en la línea 16 se lanza una excepción en caso de error, caso contrario, en la línea 18 la función callback obtiene las filas de la tabla con los datos de las especialidades médicas registradas, en la línea 23 se muestra un mensaje por consola en caso de haberse establecido un error en la conexión con la base de datos y en la línea 26 se exporta el objeto para que sea utilizado por el script que contiene las rutas del servicio. Las demás consultas SQL se encuentran definidas en el ANEXO C.


```

1  const mysql = require('mysql');
2
3  const sqlConexion = mysql.createPool({
4    host: 'localhost',
5    user: 'root',
6    password: '12u6Tsq1',
7    database: 'db_app_ciec'
8  });
9
10 let usuario = {};
11 usuario.obtenerEspecialidadesMedicas = (callback) => {
12   if(sqlConexion){
13     sqlConexion.query('SELECT * FROM tbl_Especialidad_Medica', (er
14   ror, filas) =>{
15     if (error)
16       {
17         throw error;
18       }else{
19         callback(null, filas);
20       }
21     });
22   }
23   else{
24     console.log('No se ha podido establecer conexión con la base
25   de datos');
26   }
27 }
28 module.exports = usuario;

```

Código 2.7. Conexión Base de Datos y consulta SQL obtenerEspecialidadesMedicas

Clases:

La clase Medico definida en el Código 2.8 contiene ocho atributos que representan los datos de un médico: id, apellidos, nombres, cedula, sexo, contraseña, estado e id de especialidad y corresponden a la tabla en la base de datos, estos atributos tienen la anotación `@SerializedName` que permite serializar y deserializar los objetos java de la aplicación móvil y json del servicio web. A continuación, se define un constructor con el nombre de la clase que tiene parámetros de entrada y que se emplea cuando se declara un objeto de la clase, se establece los getters y finalmente se sobrescribe el método `toString` que devuelve una cadena que incluye nombres y apellidos. Las clases restantes se encuentran definidas en el ANEXO D.

```

1  import com.google.gson.annotations.SerializedName;
2
3  public class Medico {
4      @SerializedName("id_Medico")
5      private int idMedico;
6
7      @SerializedName("apellidos_Medico")
8      private String apellidosMedico;
9
10     @SerializedName("nombres_Medico")
11     private String nombresMedico;
12
13     @SerializedName("cedula_Medico")
14     private String cedulaMedico;
15
16     @SerializedName("sexo_Medico")
17     private String sexoMedico;
18
19     @SerializedName("contrasenia_Medico")
20     private String contraseniaMedico;
21
22     @SerializedName("estado_Medico")
23     private String estadoMedico;
24
25     @SerializedName("id_Especialidad_Medica")
26     private int idEspecialidadMedico;
27
28     public Medico(String apellidosMedico, String nombresMedico, String
cedulaMedico, String sexoMedico, String contraseniaMedico, String estadoMedico,
int idEspecialidadMedico) {
29         this.apellidosMedico = apellidosMedico;
30         this.nombresMedico = nombresMedico;
31         this.cedulaMedico = cedulaMedico;
32         this.sexoMedico = sexoMedico;
33         this.contraseniaMedico = contraseniaMedico;
34         this.estadoMedico = estadoMedico;
35         this.idEspecialidadMedico = idEspecialidadMedico;
36     }
37     public int getIdMedico() {
38         return idMedico;
39     }
40     public String getApellidosMedico() {
41         return apellidosMedico;
42     }
43     public String getNombresMedico() {
44         return nombresMedico;
45     }
46     public String getCedulaMedico() {
47         return cedulaMedico;
48     }
49     public String getSexoMedico() {
50         return sexoMedico;
51     }
52     public String getContraseniaMedico() {
53         return contraseniaMedico;
54     }
55     public String getEstadoMedico() {
56         return estadoMedico;
57     }
58     public int getIdEspecialidadMedico() {
59         return idEspecialidadMedico;
60     }
61     @Override
62     public String toString() {
63         return nombresMedico + " " + apellidosMedico;
64     }
65 }

```

Código 2.8. Clase Medico

Cliente Retrofit:

En el Gradle (Código 2.9) de Android Studio se añadieron las librerías necesarias para crear un cliente HTTP para Android y emplear un convertidor en formato JSON.

```
1 implementation 'com.squareup.retrofit2:retrofit:2.6.2'
2 implementation 'com.squareup.retrofit2:converter-gson:2.6.2'
```

Código 2.9. Librerías añadidas a Gradle (Parte 1 de 2)

En el Código 2.10 se define una clase ConexionServicioWeb, la cual contiene un objeto de tipo Retrofit, la url base que contiene la dirección pública y el puerto de escucha del servicio, y un método que crea un cliente Retrofit, el cual incluye la url base y el tipo de convertidor, permitiendo al cliente acceder y consumir el servicio web.

```
1 import retrofit2.Retrofit;
2 import retrofit2.converter.gson.GsonConverterFactory;
3
4 public class ConexionServicioWeb {
5     private static Retrofit retrofit = null;
6     private static final String urlBase = "http://ec2-18-230-116-
131.sa-east-1.compute.amazonaws.com:8080";
7     public static Retrofit conexionUrl() {
8         if (retrofit == null) {
9             retrofit = new Retrofit.Builder()
10                .baseUrl(urlBase)
11                .addConverterFactory(GsonConverterFactory.create())
12                .build();
13        }
14        return retrofit;
15    }
16 }
```

Código 2.10. Creación de Cliente Retrofit

Métodos con Operaciones HTTP:

Se define una interfaz MetodoServicioWeb en la cual se declara los métodos y cada uno tiene una anotación Retrofit que representa a cada método de HTTP: @GET, @POST, @PUT, @DELETE y para cada anotación se especifica el endpoint de la url referente al servicio web. Los métodos declarados deben devolver los datos que se esperan del servidor dentro de un Call<> y estos métodos pueden tener parámetros, para ello se emplea @Path y con POST se emplea @Body. En el Código 2.11 se presentan cuatro métodos: el primero emplea @GET para obtener una lista de especialidades médicas y no tiene parámetros, el segundo emplea @POST para registrar un médico y tiene un

parámetro @Body con los datos del médico, el tercero emplea @PUT para actualizar el estado de una atención médica y tiene dos parámetros @Path con el estado e id de la atención médica y el cuarto emplea @DELETE para eliminar el registro de un médico y tiene un parámetro @Path con la cédula del médico. La interfaz MetodoServicioWeb completa se encuentra en el ANEXO D.

```
1 @GET("obtenerEspecialidadesMedicas")
2 Call<ArrayList<EspecialidadMedica>> obtenerEspecialidadesMedicas();
3
4 @POST("crearMedico")
5 Call<RespuestaCreacion> crearMedico (@Body Medico medico);
6
7 @PUT("actualizarEstadoEntregaAtencionMedica/{estado_Atencion_Medica}/{id_Atencion_Medica}")
8 Call<RespuestaActualizacion>
  actualizacionEstadoEntregaAtencionMedica(@Path("estado_Atencion_Medica"
  ) String estadoAtencionMedica, @Path("id_Atencion_Medica") int
  idAtencionMedica);
9
10 @DELETE("eliminarMedico/{cedula_Medico}")
11 Call<RespuestaEliminacion> eliminarMedico(@Path("cedula_Medico") String
  cedulaMedico);
```

Código 2.11. Declaración de Métodos con las Operaciones HTTP

2.2.3.3 CONECTIVIDAD LOCAL

Antes de emigrar los servicios a la nube, se verificó de forma local la correcta conexión entre la base de datos y el servicio web. Primero, se ejecutó el servidor local (Figura 2.42) en el puerto definido para escuchar las peticiones del cliente.

```
PS C:\Users\Administrator\Desktop\servicio-web-rest_app-ciec> node .\aplicacion\principal\app.js
Servidor en el puerto: 8080
```

Figura 2.42. Servidor Local Ejecutado en el Puerto 8080

A continuación, a través de un navegador web (Figura 2.43), se ingresó la dirección compuesta por el localhost, el puerto de escucha y la ruta de servicio, la cual permite acceder a los recursos y se obtuvo las especialidades médicas registradas en la base de datos local.

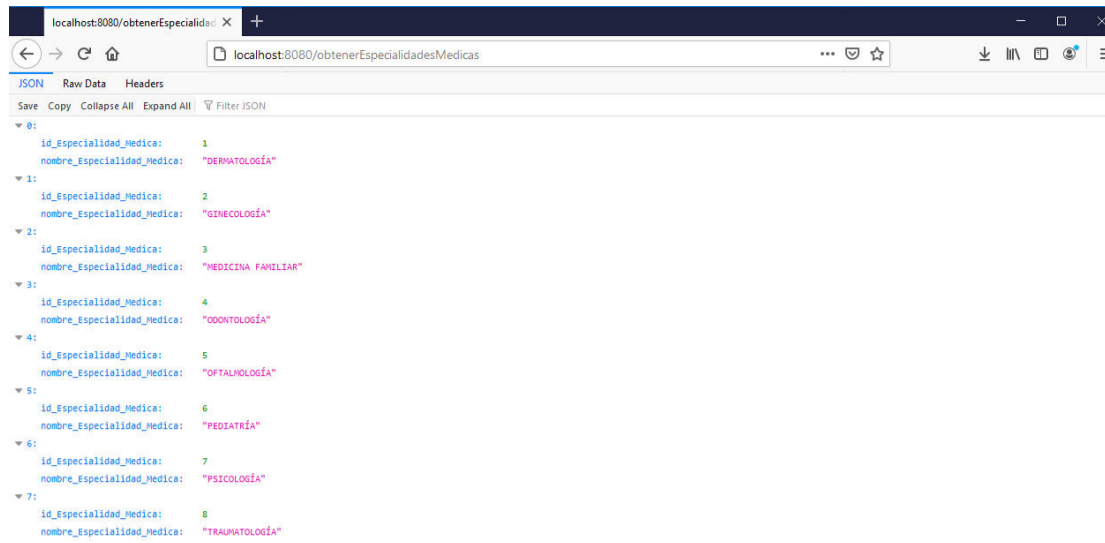


Figura 2.43. Obtención de Especialidades Médicas Registradas en Base de Datos Local a través de un Navegador Web

2.2.3.4 MIGRACIÓN DE SERVICIOS A AWS

Una vez verificada la correcta conectividad local, se realizó la migración de la base de datos al servicio RDS de AWS y del servicio web al servicio EC2 de AWS.

Migración de Base de Datos a AWS:

Para migrar la base de datos al servicio RDS de AWS, en MySQL Workbench se ingresó el hostname asignado por AWS y las credenciales: nombre de usuario y contraseña (Figura 2.44), estableciendo una conexión remota con el servicio de base de datos de la nube y se realizó la codificación similar a la Sección 2.2.3.1.

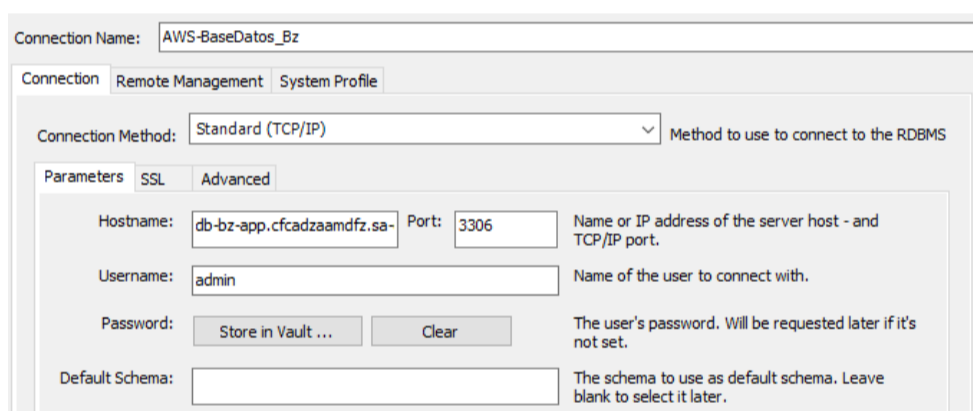


Figura 2.44. Credenciales de Acceso a Servicio de Base de Datos RDS

Migración de Servicio Web a AWS:

Para migrar el servicio web al servicio EC2 de AWS, primero se estableció la conexión con el escritorio remoto asignado por AWS (Figura 2.45).

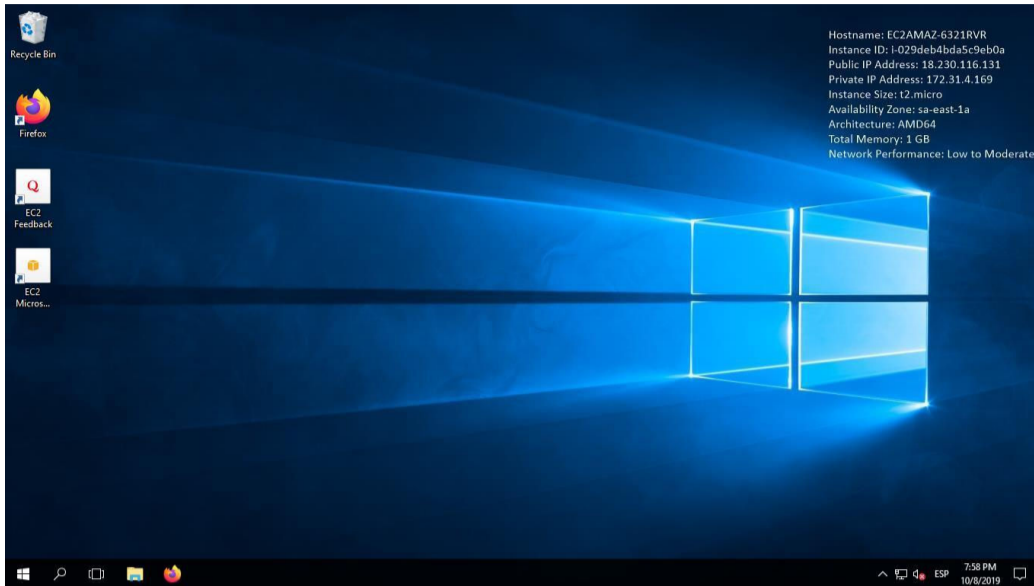


Figura 2.45. Escritorio Remoto Asignado por AWS

A continuación, se realizó el proceso de codificación similar a la Sección 2.2.3.2., con una variante en la conexión con la base de datos, que ahora se encuentra alojada en la nube de AWS. Para este proceso se especificó el host asignado por el servicio RDS de base de datos, el nombre de usuario, la contraseña y el nombre de la base de datos (Código 2.12).

```
1 const mysql = require('mysql');
2
3 const sqlConexion = mysql.createPool({
4   host: 'db-bz-app.cfcadzaamdfz.sa-east-1.rds.amazonaws.com',
5   user: 'admin',
6   password: '293wejsde8TSA',
7   database: 'db_app_ciec'
8 });
```

Código 2.12. Establecimiento de Conexión con la Base de Datos RDS de AWS

2.2.3.5 CONECTIVIDAD EN LA NUBE

Para verificar la conectividad en la nube entre la base de datos y el servicio web alojados en AWS, se ejecutó el servidor y a través de un navegador web (Figura 2.46), se ingresó la dirección compuesta por una ip pública asignada por AWS, el puerto de escucha y la

ruta de servicio, la cual permite acceder a los recursos y se obtuvo las especialidades médicas registradas en la base de datos alojada en la nube.

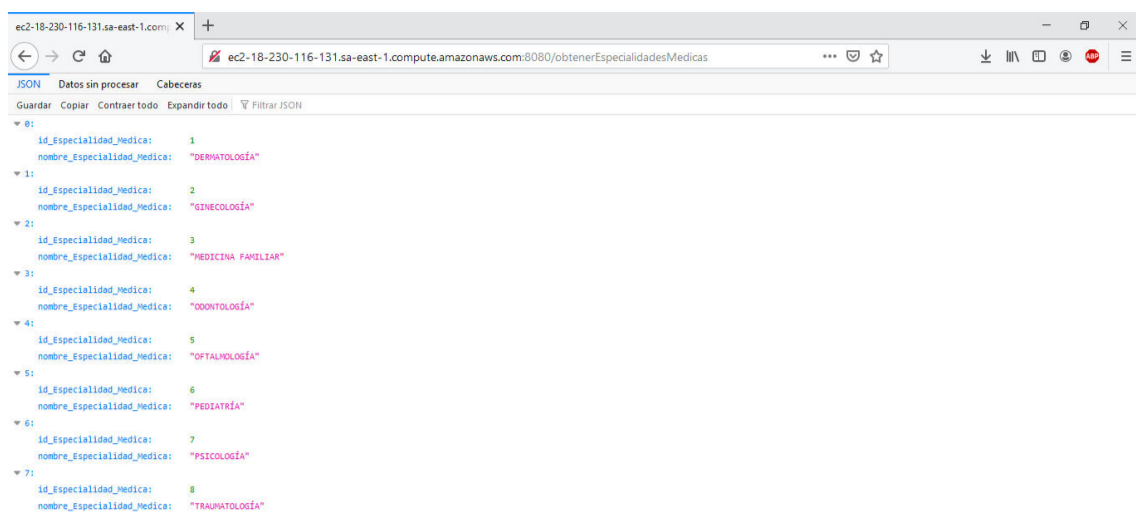


Figura 2.46. Obtención de Especialidades Médicas Registradas en Base de Datos de AWS a través de un Navegador Web

2.2.3.6 CODIFICACIÓN DE CAPA DE PRESENTACIÓN

Para la implementación de esta capa y en base al diseño realizado se codifica la interfaz gráfica de la aplicación móvil en Android Studio.

Para la creación de cada Activity en Android Studio se eligió la plantilla “Empty Activity” (Figura 2.47), es decir, que se encuentre vacía y permita implementar desde cero cada módulo de la aplicación móvil.

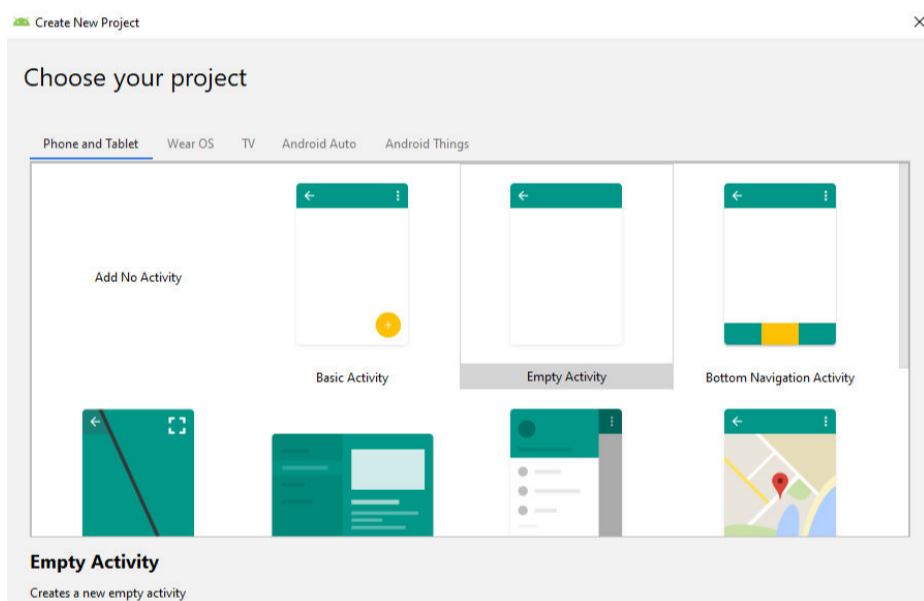


Figura 2.47. Plantillas para la creación de una Activity

La versión de Android desde la cual está disponible la aplicación móvil es Lollipop - 5.0 (Figura 2.48), para ello se ha verificado el porcentaje de dispositivos con Android (Figura 2.49) que soportan y son compatibles con esta versión. Además, el lenguaje de programación empleado es Java.

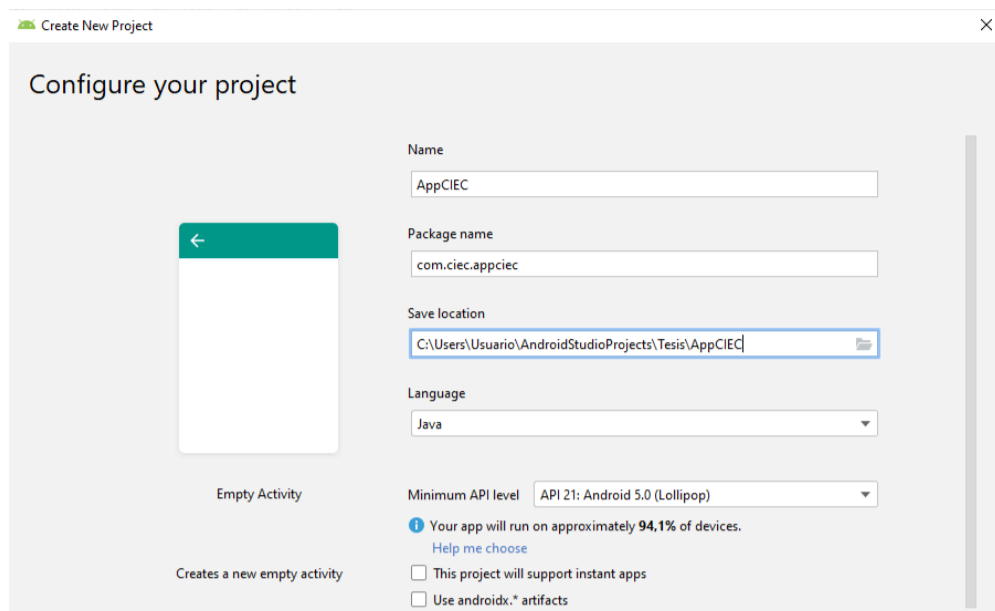


Figura 2.48. Configuración de Proyecto en Android Studio

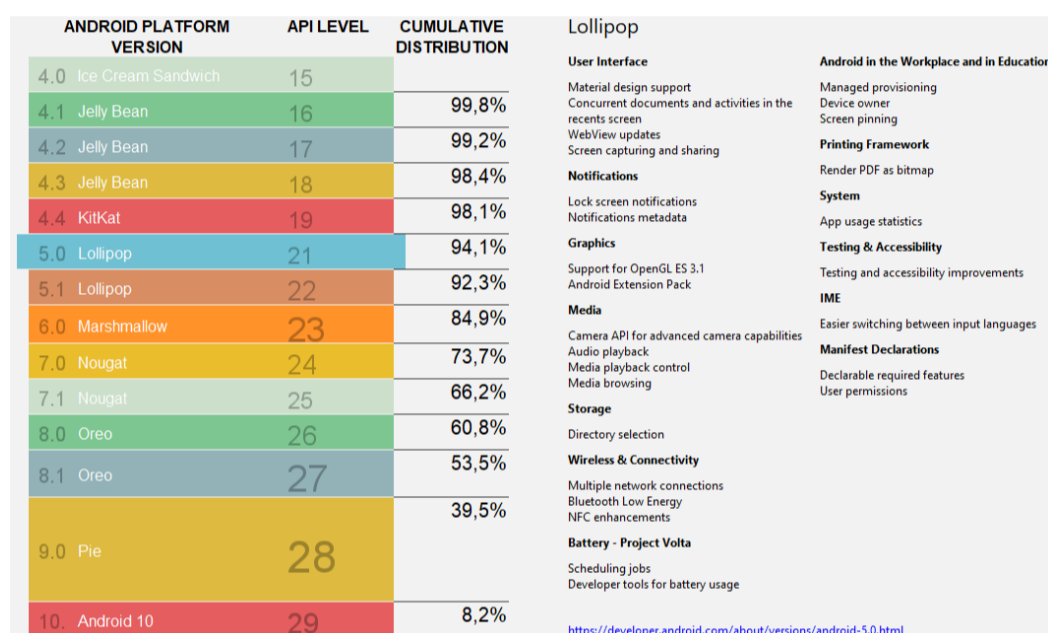


Figura 2.49. Versiones de Android

En el Gradle (Código 2.13) del proyecto se añadieron las librerías necesarias para el desarrollo de la aplicación y cada una tiene una funcionalidad distinta: la primera para

mostrar la información dentro de un diseño de tarjetas, la segunda para mostrar lista de datos, la tercera permite incluir componentes y patrones de diseño adicionales y la cuarta para crear un documento en Pdf.

```
1 implementation 'com.android.support:cardview-v7:28.0.0'  
2 implementation 'com.android.support:recyclerview-v7:28.0.0'  
3 implementation 'com.android.support:design:28.0.0'  
4 implementation 'com.itextpdf:itextg:5.5.10'
```

Código 2.13. Librerías añadidas a Gradle (Parte 2 de 2)

En el AndroidManifest del proyecto se incluyen los permisos de usuario necesarios para el funcionamiento de la aplicación, ya que, si no dispone de estos, algunas funciones de la aplicación no se pueden realizar. En el Código 2.14 se presentan dichos permisos, el primero permite acceder a Internet para consumir los servicios en la nube de AWS, el segundo y tercero permiten un almacenamiento externo de lectura y escritura a la aplicación, es decir, permiten leer y escribir en la memoria interna del dispositivo móvil.

```
1 <uses-permission android:name="android.permission.INTERNET"/>  
2 <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>  
3 <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

Código 2.14. Permisos de Usuario en el AndroidManifest

A continuación de los permisos, en el AndroidManifest se incluyen todas las Activities que contiene la aplicación, en el Código 2.15 se presentan dos, la orientación de ambas es vertical, la primera está definida entre las líneas 1 y 8, y es la Activity principal, es decir, la que se presenta al iniciar la aplicación y la segunda está definida entre las líneas 9 y 12, y contiene una propiedad para que el teclado nunca oscurezca el foco actual y los usuarios siempre puedan ver lo que están escribiendo. EL AndroidManifest completo se encuentra en el ANEXO D.

```
1 <activity  
2     android:name=".PortadaActivity"  
3     android:screenOrientation="portrait">  
4     <intent-filter>  
5         <action android:name="android.intent.action.MAIN"/>  
6         <category android:name="android.intent.category.LAUNCHER"/>  
7     </intent-filter>  
8 </activity>  
9 <activity  
10    android:name=".InicioDeSesionActivity"  
11    android:screenOrientation="portrait"  
12    android:windowSoftInputMode="adjustPan"/>
```

Código 2.15. Activity en el AndroidManifest

La codificación del Activity se ha dividido en dos partes: gráfica y lógica.

Parte Gráfica:

En los Códigos 2.16 y 2.17 se presenta parte de la implementación del Layout del Activity de Inicio de Sesión, es decir, la parte gráfica. Se ha especificado que sea LinearLayout para ordenar cada elemento de forma vertical y dos de los elementos que componen la interfaz son: el primer elemento definido entre las líneas 13 y 34 es para ingresar y modificar texto y dispone de diversas propiedades como: dimensiones de alto y ancho, margen, mensaje de error, tamaño de letra, fondo, etc., y el segundo elemento definido entre las líneas 37 y 49 es un botón que también dispone de diversas propiedades similares al anterior elemento y la propiedad onClick para ejecutar una acción definida en un método en la parte lógica del Activity.

```
1 <LinearLayout
2     xmlns:android=http://schemas.android.com/apk/res/android
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     android:orientation="vertical"
8     android:background="@drawable/fondo_inicio_de_sesion"
9     tools:context=".InicioDeSesionActivity">
10
11     ...
12
13     <android.support.design.widget.TextInputLayout
14         android:id="@+id/txtInicioSesionCedulaUsuario"
15         android:layout_width="280dp"
16         android:layout_height="wrap_content"
17         android:layout_marginTop="20dp"
18         android:layout_gravity="center_horizontal"
19         app:errorEnabled="true"
20         app:errorTextAppearance="@style/ErrorTexto">
21
22         <android.support.design.widget.TextInputEditText
23             android:layout_width="match_parent"
24             android:layout_height="45dp"
25             android:hint="@string/txt_cedula"
26             android:textSize="19sp"
27             android:inputType="number"
28             android:maxLength="10"
29             android:longClickable="false"
30             android:paddingStart="10dp"
31             android:paddingEnd="10dp"
32             android:background="@drawable/disenio_edit_text"/>
33
34     </android.support.design.widget.TextInputLayout>
```

Código 2.16. Implementación de Parte Gráfica de Activity Inicio de Sesión (Parte 1 de 2)

```

35     ...
36
37     <Button
38         android:id="@+id/btnIniciarSesionUsuario"
39         android:layout_width="175dp"
40         android:layout_height="50dp"
41         android:text="@string/btn_iniciar"
42         android:textSize="24sp"
43         android:textColor="@color/colorAzul"
44         android:fontFamily="@font/gabriela"
45         android:textAllCaps="false"
46         android:layout_marginTop="25dp"
47         android:layout_gravity="center_horizontal"
48         android:background="@drawable/disenio_boton_principal"
49         android:onClick="btnIniciarSesionUsuario"/>
50
51     ...
52 </LinearLayout>

```

Código 2.17. Implementación de Parte Gráfica de Activity Inicio de Sesión (Parte 2 de 2)

En la Figura 2.50 se presenta el resultado de la implementación del Layout Inicio de Sesión completo, en el que se incluyen todos los elementos que componen la interfaz gráfica: el primero para mostrar el texto “Inicio de Sesión”, los tres siguientes son editores de texto: Tipo de Usuario, Cédula y Contraseña, y los dos últimos son los botones: “Iniciar” y “Registrarse”. Este proceso se realiza para todas las Activities de acuerdo a los elementos requeridos en la parte gráfica.



Figura 2.50. Interfaz Gráfica de Activity Inicio de Sesión

Parte Lógica:

En el Código 2.18 se ha implementado la validación de la Cédula de Identidad, para ello, se ha definido una clase que contiene un método para dicho proceso, en el cual se establece que los dos primeros dígitos deben estar en el rango de 01 a 24, excluyendo el 00, debido al número de provincias. Se define un coeficiente verificador por el que se debe multiplicar los nueve primeros dígitos y el décimo dígito es el valor que permite verificar que la cédula sea válida o no.

```
1 public class Cedula {
2
3     public boolean validarCedula(String cedula){
4
5         int primerDigito = Integer.parseInt(cedula.substring(0,1));
6         int segundoDigito = Integer.parseInt(cedula.substring(1,2));
7         int[] coeficienteVerificador = { 2, 1, 2, 1, 2, 1, 2, 1, 2 };
8         int verificador = Integer.parseInt(cedula.substring(9,10));
9         int suma = 0;
10        int digito = 0;
11        int resultado = 0;
12
13        if(primerDigito >= 0 && primerDigito <= 2){
14
15            if(primerDigito == 0 && segundoDigito == 0){
16                return false;
17            }
18            else if (primerDigito == 2 && segundoDigito > 4){
19                return false;
20            } else {
21                for (int i = 0; i < (cedula.length() - 1); i++) {
22
23                    digito = Integer.parseInt(cedula.substring(i, i + 1))*
coeficienteVerificador[i];
24                    suma += ((digito % 10) + (digito / 10));
25                }
26                resultado = suma % 10;
27
28                if(resultado == 0 && resultado == verificador){
29                    return true;
30                }
31                else if ((10 - resultado) == verificador){
32                    return true;
33                }
34                else {
35                    return false;
36                }
37            }
38        }
39        else{
40            return false;
41        }
42    }
43 }
```

Código 2.18. Validación de Cédula de Identidad

En el Código 2.19 se define el método que permite realizar el registro de un usuario médico consumiendo el servicio web, para ello: se establece la conexión con el servicio

web invocando al método de la clase ConexionServicioWeb (Código 2.10) que contiene dicha conexión, se instancia un objeto de la clase Medico (Código 2.8) con los datos a registrarse, se invoca al método definido en la interfaz MetodoServicioWeb (Código 2.11) que envía una solicitud al servidor para obtener una respuesta y se realiza una llamada asíncrona, la respuesta puede ser exitosa o no, o se puede obtener un fallo en la conexión. En el ANEXO D se encuentra la codificación de la aplicación móvil.

```

1 private void registrarUsuarioMedico(){
2     MetodoServicioWeb metodoServicioWeb =
3     ConexionServicioWeb.conexionUrl().create(MetodoServicioWeb.class);
4     Medico medico = new
5     Medico(apellidos,nombres,cedula,sexo,contrasenia,"Activo",
6     idEspecialidad);
7
8     Call<RespuestaCreacion> callCrearMedico =
9     metodoServicioWeb.crearMedico(medico);
10    callCrearMedico.enqueue(new Callback<RespuestaCreacion>() {
11        @Override
12        public void onResponse(Call<RespuestaCreacion> call,
13        Response<RespuestaCreacion> response) {
14
15            if(response.isSuccessful()){
16                Toast.makeText(getApplicationContext(), "Médico
17                Registrado Correctamente", Toast.LENGTH_LONG).show();
18            } else{
19                switch (response.code()) {
20                    case 404:
21                        Toast.makeText(getApplicationContext(),
22                        "Error al registrar los datos en el servidor",
23                        Toast.LENGTH_SHORT).show();
24                        break;
25                    case 500:
26                        Toast.makeText(getApplicationContext(), "Se
27                        produjo un error en el servidor", Toast.LENGTH_SHORT).show();
28                        break;
29                    default:
30                        Toast.makeText(getApplicationContext(),
31                        "Error desconocido", Toast.LENGTH_SHORT).show();
32                        break;
33                }
34            }
35        }
36        @Override
37        public void onFailure(Call<RespuestaCreacion> call,
38        Throwable t) {
39            Toast.makeText(getApplicationContext(),"No se ha podido
40            conectar con el servidor",Toast.LENGTH_SHORT).show();
41        }
42    });
43 }

```

Código 2.19. Método de Registro de Usuario Médico

3 RESULTADOS Y DISCUSIÓN

En este capítulo primero se realiza la actualización del tablero Kanban y a continuación, para comprobar el cumplimiento de todos los requerimientos funcionales se presentan las pruebas de funcionamiento del prototipo y encuestas de validación de resultados a los usuarios de la aplicación móvil.

3.1 ACTUALIZACIÓN DEL TABLERO KANBAN

En la actualización de actividades del tablero Kanban (Tabla 3.1), las tareas que se antes se encontraban en proceso de desarrollo han sido culminadas, por lo que ahora se encuentran incluidas en la columna de tareas realizadas y las tareas que estaban pendientes, ahora se encuentran en proceso de desarrollo.

Tabla 3.1. Segunda Actualización de Tablero Kanban

Tareas Pendientes	Tareas en Proceso	Tareas Realizadas
	Pruebas de funcionamiento del prototipo	Definición de conceptos teóricos
	Encuestas de validación de funcionamiento a usuarios	Entrevistas a administrador, personal médico y dependiente de farmacia del centro de salud
		Identificación de Requerimientos Funcionales
		Identificación de Requerimientos No Funcionales
		Definición de módulos de la aplicación móvil
		Diseño de Capa de Datos
		Diseño de Capa de Negocio
		Diseño de Capa de Presentación
		Creación de una instancia de EC2 con Windows Server
		Creación de una instancia de RDS con MySql
		Instalación de Node.js y Visual Studio Code en Windows Server
		Instalación de Android Studio
		Codificación de Capa de Datos
		Codificación de Capa de Negocio
		Codificación de Capa de Presentación

3.2 PRUEBAS DE FUNCIONAMIENTO

Las pruebas de funcionamiento del prototipo se han realizado para validar el cumplimiento de los requerimientos funcionales descritos en el Apartado 2.1.2.4.

3.2.1 PRUEBAS DE VALIDACIÓN DE REQUERIMIENTOS FUNCIONALES: MODULO REGISTRO DE USUARIO

3.2.1.1 REQUERIMIENTO FUNCIONAL RF01: CREAR NUEVO USUARIO

- Administrador:

El usuario administrador para registrarse ingresa sus datos personales: apellidos, nombres, cédula, selecciona el sexo, escoge el tipo de usuario, en este caso Administrador e ingresa una contraseña y su confirmación. El botón “Registrarse” realiza el proceso de registro en la base de datos (Figura 3.1). Sólo existe un Administrador Registrado Activo

Registro De Usuario

Apellidos: Urbina Mena

Nombres: David Felipe

Cédula: 1725854754

Sexo: Femenino Masculino

Tipo de Usuario: ADMINISTRADOR

Contraseña: AQ78@fr

Confirmar Contraseña: AQ78@fr

Registrarse

Figura 3.1. Registro de Usuario Administrador

En la tabla tbl_Administrador de la base de datos se verifica el registro del usuario Administrador con los datos ingresados desde la aplicación móvil (Figura 3.2).

id_Administrador	apellidos_Administrador	nombres_Administrador	cedula_Administrador	sexo_Administrador	contrasenia_Administrador	estado_Administrador
1	Urbina Mena	David Felipe	1725854754	Masculino	AQ78@fr	Activo

Figura 3.2. Nuevo Usuario Administrador Registrado en la Base De Datos

- Médico:

El usuario médico para registrarse ingresa sus datos personales: apellidos, nombres, cédula, selecciona el sexo, escoge el tipo de usuario, en este caso Médico, selecciona la especialidad e ingresa una contraseña y su confirmación. El botón “Registrarse” realiza el proceso de registro en la base de datos (Figura 3.3).

Figura 3.3. Registro de Usuario Médico

En la tabla tbl_Médico de la base de datos se verifica el registro del usuario Médico con los datos ingresados desde la aplicación móvil y con el estado “Pendiente” debido a la espera de la validación del registro por parte del administrador (Figura 3.4).

id_Médico	apellidos_Médico	nombres_Médico	cedula_Médico	sexo_Médico	contrasenia_Médico	estado_Médico	id_Especialidad_Médica
1	Medina Portilla	Lorena Tatiana	1400633010	Femenino	45Qer\$77	Pendiente	5
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figura 3.4. Nuevo Usuario Médico Registrado en la Base De Datos

Para la validación del registro del médico, el administrador selecciona el tipo de usuario, en este caso Médico y se despliega una lista con el usuario registrado que tiene un estado “Pendiente” (Figura 3.5).



Figura 3.5. Usuario Médico Registrado Pendiente de Validación

A continuación, el Administrador selecciona al médico y se despliega un cuadro de dialogo que contiene varias opciones y elige la opción "Activar" (Figura 3.6).

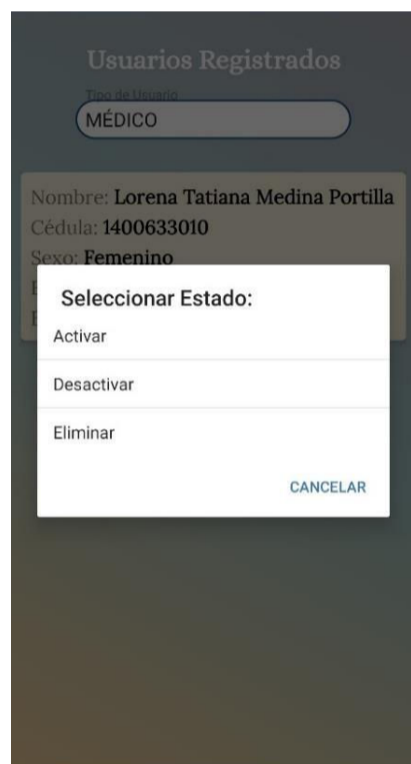


Figura 3.6. Cuadro de Dialogo de Activación de Usuario Médico

Finalmente, la lista se actualiza y el médico ya tiene un estado de “Activo” y puede acceder las funcionalidades correspondientes de la aplicación (Figura 3.7).



Figura 3.7. Usuario Médico Activado

En la tabla tbl_Médico de la base de datos se actualiza el estado del usuario médico a “Activo” después de la validación por parte del administrador (Figura 3.8).

id_Médico	apellidos_Médico	nombres_Médico	cedula_Médico	sexo_Médico	contrasenia_Médico	estado_Médico	id_Especialidad_Médico
1	Medina Portilla	Lorena Tatiana	1400633010	Femenino	45Qer\$77	Activo	5

Figura 3.8. Actualización de Estado de Médico a “Activo” en la tabla tbl_Médico en la Base de Datos

- Dependiente de Farmacia:

El usuario dependiente de farmacia para registrarse ingresa sus datos personales: apellidos, nombres, cédula, selecciona el sexo, escoge el tipo de usuario, en este caso Dependiente de Farmacia e ingresa una contraseña y su confirmación. El botón “Registrarse” realiza el proceso de registro en la base de datos (Figura 3.9).

Registro De Usuario

Apellidos
Lara Artiaga

Nombres
María Fernanda

Cédula
1804025318

Sexo: Femenino Masculino

Tipo de Usuario
DEPENDIENTE DE FARMACIA

Contraseña
Tus63a#

Mayúsculas, minúsculas, números, @#%\$!*

Confirmar Contraseña
Tus63a#

Registrarse

Figura 3.9. Registro de Usuario Dependiente de Farmacia

En la tabla tbl_Dependiente_Farmacia de la base de datos se verifica el registro del usuario dependiente de farmacia con los datos ingresados desde la aplicación móvil y con el estado "Pendiente" debido a la espera de la validación del registro por parte del administrador (Figura 3.10).

id_Dependiente_Far	apellidos_Dependiente_Farm	nombres_Dependiente_Farri	cedula_Dependiente_Farm	sexo_Dependiente_Fa	contrasenia_Dependiente_Far	estado_Dependiente_Farri
1	Lara Artiaga	María Fernanda	1804025318	Femenino	Tus63a#	Pendiente

Figura 3.10. Nuevo Usuario Dependiente de Farmacia Registrado en la Base De Datos

Para la validación del registro del dependiente de farmacia, el administrador selecciona el tipo de usuario, en este caso Dependiente de Farmacia y se despliega una lista con el usuario registrado que tiene un estado de Pendiente de activación (Figura 3.11)



Figura 3.11. Usuario Dependiente de Farmacia Registrado Pendiente de Validación

A continuación, el Administrador selecciona al dependiente de farmacia y se despliega un cuadro de dialogo que contiene varias opciones y elige la opción "Activar" (Figura 3.12).

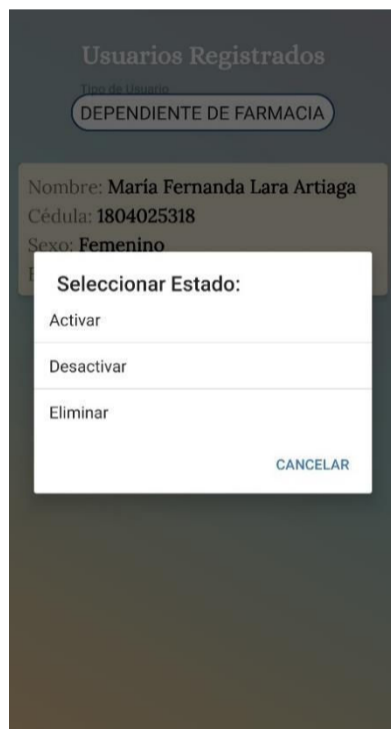


Figura 3.12. Cuadro de Dialogo de Activación de Usuario Dependiente de Farmacia

Finalmente, la lista se actualiza y el dependiente de farmacia ya tiene un estado de “Activo” y puede acceder las funcionalidades correspondientes de la aplicación (Figura 3.13).



Figura 3.13. Usuario Dependiente de Farmacia Activado

En la tabla tbl_Dependiente_Farmacia de la base de datos se actualiza el estado del usuario dependiente de farmacia a “Activo” después de la validación por parte del administrador (Figura 3.14).

id_Dependiente_Far	apellidos_Dependiente_Farm	nombres_Dependiente_Farri	cedula_Dependiente_Farm	sexo_Dependiente_Fa	contrasenia_Dependiente_Far	estado_Dependiente_Farm
1	Lara Artiaga	María Fernanda	1804025318	Femenino	Tus63a#	Activo

Figura 3.14. Actualización de Estado de Dependiente de Farmacia a “Activo” en la tabla tbl_Dependiente_Farmacia en la Base de Datos

3.2.2 PRUEBAS DE VALIDACIÓN DE REQUERIMIENTOS FUNCIONALES: MODULO INICIO DE SESIÓN DE USUARIO

3.2.2.1 REQUERIMIENTO FUNCIONAL RF02: INGRESAR CREDENCIALES DE USUARIO

- Administrador:

El usuario administrador registrado previamente, para iniciar sesión selecciona el tipo de usuario Administrador e ingresa su cédula y contraseña, y el botón “Iniciar” (Figura 3.15) permite acceder al interfaz principal del usuario.



Figura 3.15. Inicio de Sesión de Usuario Administrador

Una vez realizado el proceso anterior, accede al interfaz principal del usuario Administrador y en la parte inferior se visualiza el mensaje "Inicio de sesión de Administrador correcto" (Figura 3.16).



Figura 3.16. Mensaje de Inicio de Sesión de Usuario Administrador Correcto

- Médico:

El usuario médico registrado previamente, para iniciar sesión selecciona el tipo de usuario Médico e ingresa su cédula y contraseña, y el botón “Iniciar” (Figura 3.17) permite acceder al interfaz principal del usuario.



Figura 3.17. Inicio de Sesión de Usuario Médico

Una vez realizado el proceso anterior, accede al interfaz principal del usuario Médico y en la parte inferior se visualiza el mensaje “Inicio de sesión de Médico correcto” (Figura 3.18).

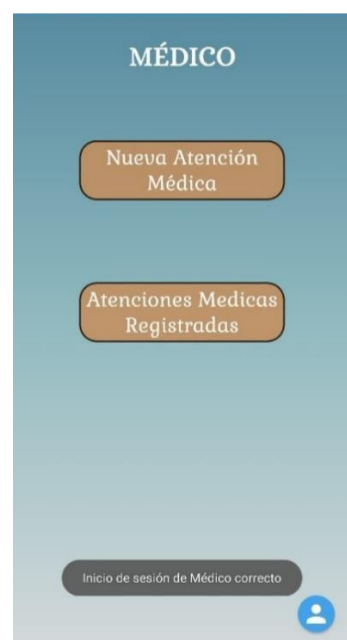


Figura 3.18. Mensaje de Inicio de Sesión de Usuario Médico Correcto

- Dependiente de Farmacia:

El usuario dependiente de farmacia registrado previamente, para iniciar sesión selecciona el tipo de usuario Dependiente de Farmacia e ingresa su cédula y contraseña, y el botón “Iniciar” (Figura 3.19) permite acceder al interfaz principal del usuario.



Figura 3.19. Inicio de Sesión de Usuario Dependiente de Farmacia

Una vez realizado el proceso anterior, accede al interfaz principal del usuario Dependiente de Farmacia y en la parte inferior se visualiza el mensaje “Inicio de sesión de Dependiente de Farmacia correcto” (Figura 3.20).



Figura 3.20. Mensaje de Inicio de Sesión de Usuario Dependiente de Farmacia Correcto

3.2.3 PRUEBAS DE VALIDACIÓN DE REQUERIMIENTOS FUNCIONALES: MÓDULO DATOS PERSONALES DE USUARIO

3.2.3.1 REQUERIMIENTO FUNCIONAL RF03: VISUALIZAR DATOS DE USUARIO

Se ha creado un nuevo usuario tipo Médico repitiendo el proceso para RF01. En la base de datos se verifica al segundo usuario registrado en la tabla tbl_Medico (Figura 3.21).

id_Medico	apellidos_Medico	nombres_Medico	cedula_Medico	sexo_Medico	contrasenia_Medico	estado_Medico	id_Especialidad_Medica
1	Medina Portilla	Lorena Tatiana	1400633010	Femenino	45Qer\$77	Activo	5
2	Álvarez Herrera	Henry Renato	1718026790	Masculino	gcn%9107E	Activo	8

Figura 3.21. Segundo Usuario Médico Registrado en la Base De Datos

El usuario médico visualiza sus datos registrados: apellidos, nombres, especialidad, cédula y sexo (Figura 3.22).

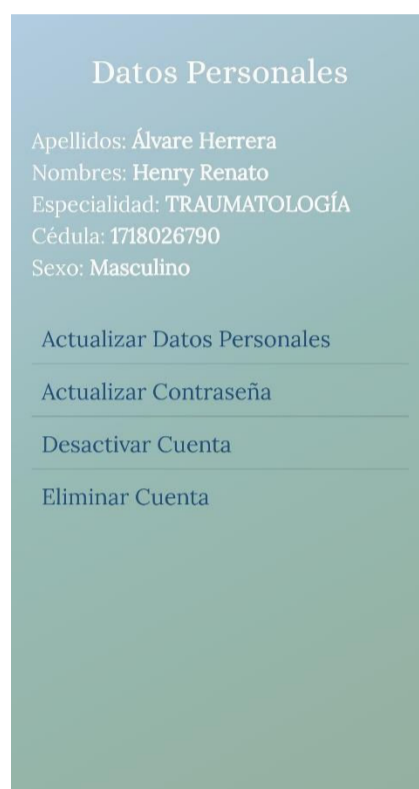


Figura 3.22. Médico Visualiza sus Datos Personales

3.2.3.2 REQUERIMIENTO FUNCIONAL RF04: ACTUALIZAR DATOS DE USUARIO

El médico actualiza su primer apellido, registrado previamente con un error ortográfico (Figura 3.23).

Figura 3.23. Médico Actualiza sus Datos Personales

En la base de datos se verifica la actualización del primer apellido del usuario en la tabla tbl_Medico (Figura 3.24).

	id_Medico	apellidos_Medico	nombres_Medico	cedula_Medico	sexo_Medico	contrasenia_Medico	estado_Medico	id_Especialidad_Medica
▶	1	Medina Portilla	Lorena Tatiana	1400633010	Femenino	45Qer\$77	Activo	5
	2	Álvarez Herrera	Henry Renato	1718026790	Masculino	gcn%9107E	Activo	8
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figura 3.24. Actualización de Apellido en la tabla tbl_Medico en la Base de Datos

3.2.3.3 REQUERIMIENTO FUNCIONAL RF05: ACTUALIZAR CONTRASEÑA DE USUARIO

El médico actualiza su contraseña, haciéndola más extensa para incrementar la seguridad (Figura 3.25).



Figura 3.25. Médico Actualiza su Contraseña

En la base de datos se verifica la actualización de la contraseña del usuario en la tabla tbl_Medico (Figura 3.26).

id_Medico	apellidos_Medico	nombres_Medico	cedula_Medico	sexo_Medico	contrasenia_Medico	estado_Medico	id_Especialidad_Medica
1	Medina Portilla	Lorena Tatiana	1400633010	Femenino	45Qer\$77	Activo	5
2	Álvarez Herrera	Henry Renato	1718026790	Masculino	gcn%9107E28d	Activo	8
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figura 3.26. Actualización de Contraseña en la tabla tbl_Medico en la Base de Datos

3.2.3.4 REQUERIMIENTO FUNCIONAL RF06: DESACTIVAR USUARIO

El usuario médico al desactivar su cuenta, mantiene sus datos registrados, pero ya no tiene acceso a las funcionalidades de la aplicación (Figura 3.27).

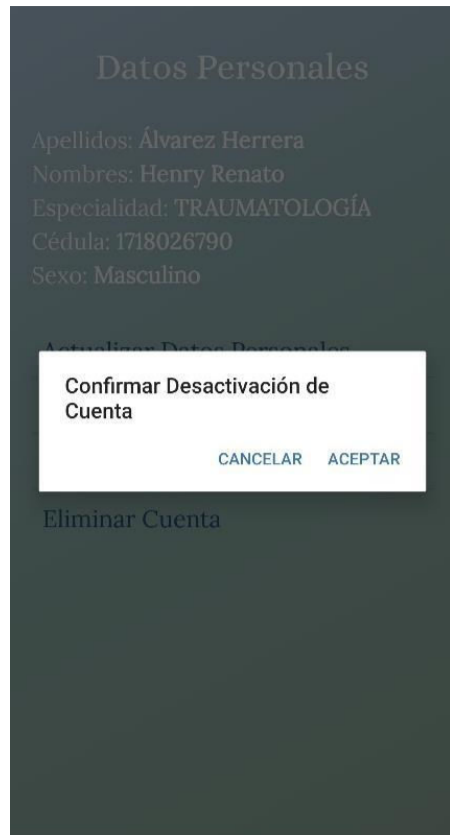


Figura 3.27. Médico Desactiva su Cuenta

En la base de datos se verifica la actualización del estado del usuario a “Desactivo” en la tabla tbl_Medico (Figura 3.28).

	id_Medico	apellidos_Medico	nombres_Medico	cedula_Medico	sexo_Medico	contrasenia_Medico	estado_Medico	id_Especialidad_Medica
▶	1	Medina Portilla	Lorena Tatiana	1400633010	Femenino	45Qer\$77	Activo	5
	2	Álvarez Herrera	Henry Renato	1718026790	Masculino	gcn%9107E28d	Desactivo	8
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figura 3.28. Actualización de Estado de Usuario a Desactivo en la tabla tbl_Medico en la Base de Datos

3.2.3.5 REQUERIMIENTO FUNCIONAL RF07: ELIMINAR USUARIO

El usuario médico al eliminar su cuenta, sus datos registrados son borrados de forma permanente (Figura 3.29).

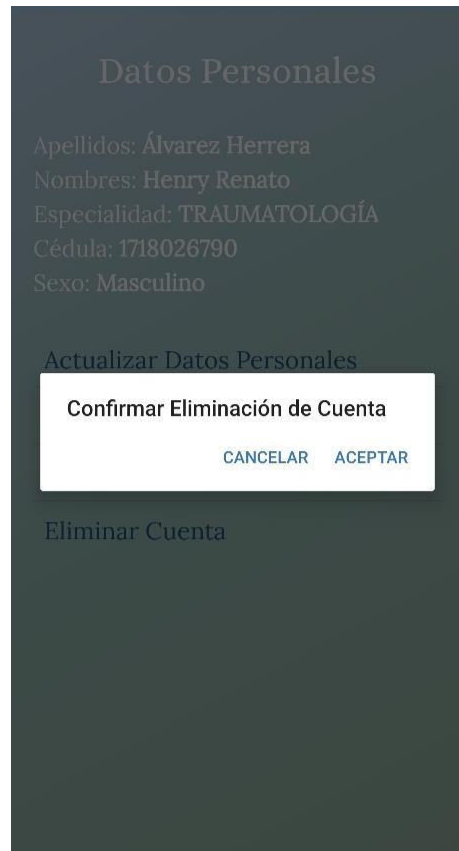


Figura 3.29. Médico Elimina su Cuenta

En la base de datos se verifica la eliminación del usuario en la tabla tbl_Medico (Figura 3.30).

	id_Medico	apellidos_Medico	nombres_Medico	cedula_Medico	sexo_Medico	contrasenia_Medico	estado_Medico	id_Especialidad_Medica
▶	1	Medina Portilla	Lorena Tatiana	1400633010	Femenino	45Qer\$77	Activo	5
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figura 3.30. Eliminación de Médico en la tabla tbl_Medico en la Base de Datos

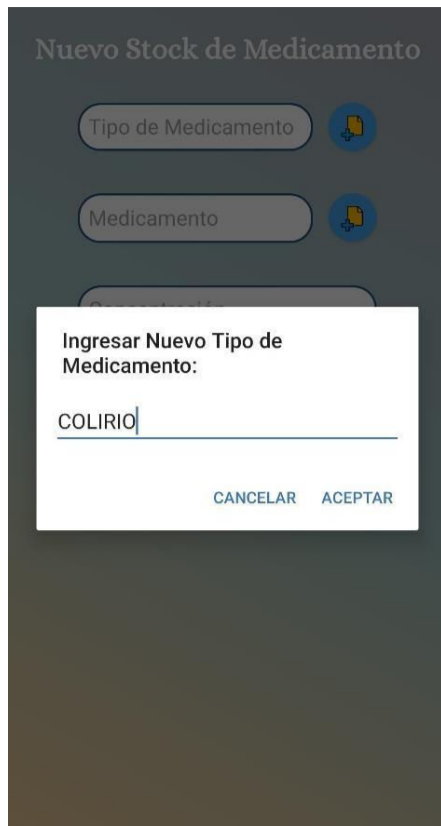
3.2.4 PRUEBAS DE VALIDACIÓN DE REQUERIMIENTOS FUNCIONALES: MÓDULO NUEVO STOCK DE MEDICAMENTO

3.2.4.1 REQUERIMIENTO FUNCIONAL RF08: INGRESAR DATOS DE UN MEDICAMENTO

Para ingresar los datos de un nuevo medicamento (Figura 3.31), el administrador primero agrega un nuevo tipo de medicamento. Para este proceso, selecciona el botón circular ubicado en la parte derecha del campo Tipo de Medicamento (Parte a) y se despliega un cuadro de dialogo que contiene un campo (Parte b), donde ingresa el nuevo tipo de medicamento (Parte c).



(a)



(b)



(c)

Figura 3.31. Ingreso de Nuevo Tipo de Medicamento

A continuación, el administrador selecciona el botón circular ubicado en la parte derecha del campo Medicamento (Figura 3.32) y se despliega un cuadro de dialogo que contiene dos campos (Parte a), donde ingresa los datos de un nuevo medicamento y su concentración (Parte b).

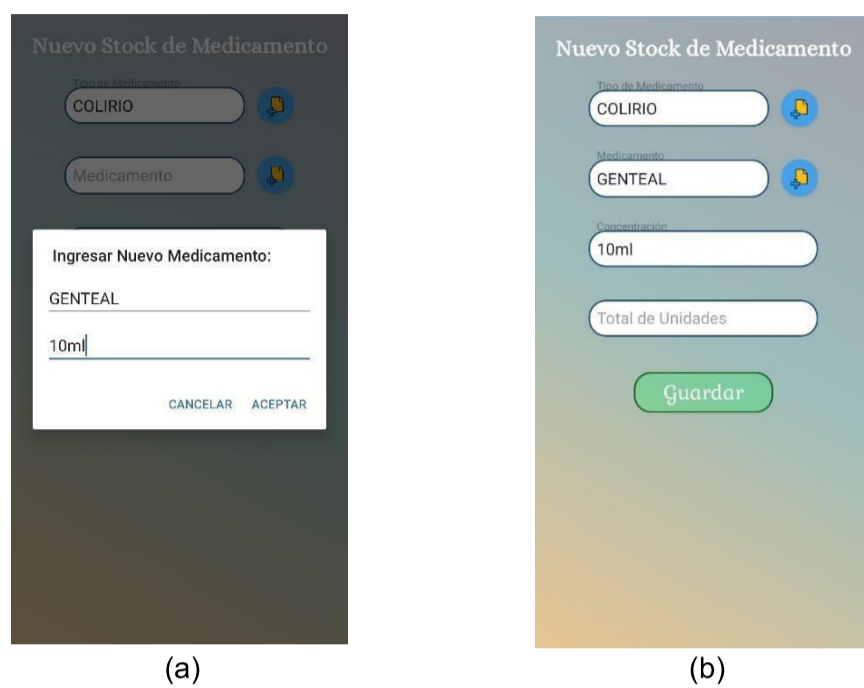


Figura 3.32. Ingreso de Nuevo Medicamento y su Concentración

Finalmente, en el último campo de texto el administrador ingresa la cantidad total de unidades de dicho medicamento (Figura 3.33).



Figura 3.33. Ingreso de Unidades de Nuevo Medicamento

3.2.4.2 REQUERIMIENTO FUNCIONAL RF09: CREAR STOCK DE MEDICAMENTO

El botón “Guardar” permite crear el stock del medicamento y retorna al interfaz principal del Administrador donde en la parte inferior se visualiza el mensaje “Nuevo Stock Registrado con Éxito” (Figura 3.34).



Figura 3.34. Creación de Stock de Medicamento

En la base de datos se verifica el registro del nuevo tipo de medicamento en la tabla tbl_Tipo_Medicamento (Figura 3.35).

	id_Tipo_Medicamento	nombre_Tipo_Medicamento
▶	1	COLIRIO
*	NULL	NULL

Figura 3.35. Nuevo Tipo de Medicamento Registrado en la Base De Datos

En la base de datos se verifica el registro del nuevo medicamento en la tabla tbl_Medicamento (Figura 3.36).

	id_Medicamento	nombre_Medicamento	concentracion_Medicamento	id_Tipo_Medicamento
▶	1	GENTEAL	10ml	1
*	NULL	NULL	NULL	NULL

Figura 3.36. Nuevo Medicamento Registrado en la Base De Datos

En la base de datos se verifica el registro del nuevo stock de medicamento en la tabla tbl_Medicamento_Stock (Figura 3.37).

id_Medicamento_Stock	total_Unidades_Ingreso_Medicamento_Stock	saldo_Unidades_Medicamento_Stock	fecha_Ingreso_Medicamento_Stock	id_Medicamento	id_Administrador
1	50	50	2020-10-12	1	1
NULL	NULL	NULL	NULL	NULL	NULL

Figura 3.37. Nuevo Stock de Medicamento Registrado en la Base De Datos

Para crear más stocks de medicamentos, el administrador repite los procesos seguidos para los requerimientos funcionales RF08 y RF09.

En la base de datos se verifica el registro de los cuatro nuevos tipos de medicamentos en la tabla tbl_Tipo_Medicamento (Figura 3.38).

id_Tipo_Medicamento	nombre_Tipo_Medicamento
1	COLIRIO
2	CREMA
3	GOTAS
4	JARABE
5	TABLETA
NULL	NULL

Figura 3.38. Cuatro Nuevos Tipos de Medicamentos Registrados en la Base De Datos

En la base de datos se verifica el registro de nueve nuevos medicamentos en la tabla tbl_Medicamento (Figura 3.39).

id_Medicamento	nombre_Medicamento	concentracion_Medicamento	id_Tipo_Medicamento
1	GENTEAL	10ml	1
2	CANESTEN	20g	2
3	ANTIBIOX	30ml	3
4	VAPORAL	120ml	4
5	DICLOFENACO	50mg	5
6	TACROLIMUS	15g	2
7	SYSTANE ULTRA	10ml	1
8	CLARITROMICINA	50ml	4
9	PARACETAMOL	80mg	5
10	OMEPRAZOL	40mg	5

Figura 3.39. Nueve Nuevos Medicamentos Registrados en la Base De Datos

En la base de datos se verifica el registro de nueve nuevos stocks de medicamentos en la tabla tbl_Medicamento_Stock (Figura 3.40).

id_Medicamento_Stock	total_Unidades_Ingreso_Medicamento_Stock	saldo_Unidades_Medicamento_Stock	fecha_Ingreso_Medicamento_Stock	id_Medicamento	id_Administrador
1	50	50	2020-10-12	1	1
2	25	25	2020-10-12	2	1
3	20	20	2020-10-12	3	1
4	5	5	2020-10-12	4	1
5	45	45	2020-10-12	5	1
6	10	10	2020-10-12	6	1
7	20	20	2020-10-12	7	1
8	30	30	2020-10-12	8	1
9	60	60	2020-10-12	9	1
10	75	75	2020-10-12	10	1

Figura 3.40. Nueve Nuevos Stocks de Medicamentos Registrados en la Base De Datos

3.2.5 PRUEBAS DE VALIDACIÓN DE REQUERIMIENTOS FUNCIONALES: MÓDULO LISTA DE MEDICAMENTOS EN STOCK

3.2.5.1 REQUERIMIENTO FUNCIONAL RF10: VISUALIZAR MEDICAMENTOS EN STOCK

El administrador visualiza la lista de stocks de medicamentos que ha registrado previamente (Figura 3.41).



Figura 3.41. Lista de Medicamentos en Stock Registrados

3.2.5.2 REQUERIMIENTO FUNCIONAL RF11: ELIMINAR MEDICAMENTO EN STOCK

En caso de error en el registro del total de unidades de uno de los stocks, el administrador elimina dicho stock de la lista. Para este proceso, selecciona el stock y se despliega un cuadro de dialogo donde confirma la eliminación del mismo. (Figura 3.42).

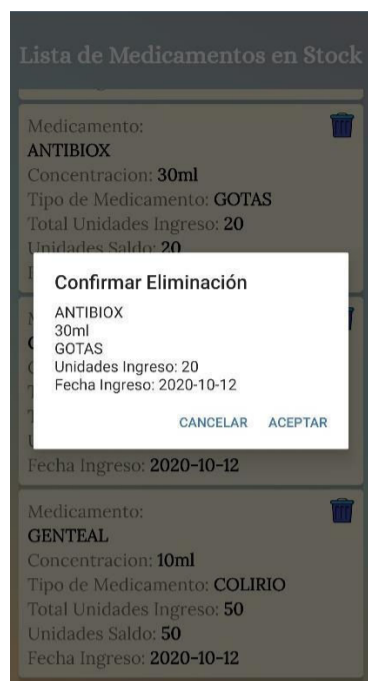


Figura 3.42. Confirmación de Eliminación de Stock de Medicamento

En la base de datos se verifica la eliminación del stock de medicamento en la tabla tbl_Medicamento_Stock (Figura 3.43).

id_Medicamento_Stock	total_Unidades_Ingreso_Medicamento_Stock	saldo_Unidades_Medicamento_Stock	fecha_Ingreso_Medicamento_Stock	id_Medicamento	id_Administrador
1	50	50	2020-10-12	1	1
2	25	25	2020-10-12	2	1
4	5	5	2020-10-12	4	1
5	45	45	2020-10-12	5	1
6	10	10	2020-10-12	6	1
7	20	20	2020-10-12	7	1
8	30	30	2020-10-12	8	1
9	60	60	2020-10-12	9	1
10	75	75	2020-10-12	10	1

Figura 3.43. Eliminación de Stock de Medicamento en la tabla tbl_Medicamento_Stock en la Base de Datos




A continuación, la lista de stocks de medicamentos se actualiza, sin el stock que ha sido eliminado. (Figura 3.44)

Lista de Medicamentos en Stock

<p>Medicamento: </p> <p>VAPORAL</p> <p>Concentracion: 120ml</p> <p>Tipo de Medicamento: JARABE</p> <p>Total Unidades Ingreso: 5</p> <p>Unidades Saldo: 5</p> <p>Fecha Ingreso: 2020-10-12</p>
<p>Medicamento: </p> <p>CANESTEN</p> <p>Concentracion: 20g</p> <p>Tipo de Medicamento: CREMA</p> <p>Total Unidades Ingreso: 25</p> <p>Unidades Saldo: 25</p> <p>Fecha Ingreso: 2020-10-12</p>
<p>Medicamento: </p> <p>GENTEAL</p> <p>Concentracion: 10ml</p> <p>Tipo de Medicamento: COLIRIO</p> <p>Total Unidades Ingreso: 50</p> <p>Unidades Saldo: 50</p> <p>Fecha Ingreso: 2020-10-12</p>

(a)

Lista de Medicamentos en Stock

<p>Medicamento: </p> <p>SYSTANE ULTRA</p> <p>Concentracion: 10ml</p> <p>Tipo de Medicamento: COLIRIO</p> <p>Total Unidades Ingreso: 20</p> <p>Unidades Saldo: 20</p> <p>Fecha Ingreso: 2020-10-12</p>
<p>Medicamento: </p> <p>TACROLIMUS</p> <p>Concentracion: 15g</p> <p>Tipo de Medicamento: CREMA</p> <p>Total Unidades Ingreso: 10</p> <p>Unidades Saldo: 10</p> <p>Fecha Ingreso: 2020-10-12</p>
<p>Medicamento: </p> <p>DICLOFENACO</p> <p>Concentracion: 50mg</p> <p>Tipo de Medicamento: TABLETA</p> <p>Total Unidades Ingreso: 45</p> <p>Unidades Saldo: 45</p> <p>Fecha Ingreso: 2020-10-12</p>

(b)

Lista de Medicamentos en Stock

<p>Medicamento: </p> <p>OMEPRAZOL</p> <p>Concentracion: 40mg</p> <p>Tipo de Medicamento: TABLETA</p> <p>Total Unidades Ingreso: 75</p> <p>Unidades Saldo: 75</p> <p>Fecha Ingreso: 2020-10-12</p>
<p>Medicamento: </p> <p>PARACETAMOL</p> <p>Concentracion: 80mg</p> <p>Tipo de Medicamento: TABLETA</p> <p>Total Unidades Ingreso: 60</p> <p>Unidades Saldo: 60</p> <p>Fecha Ingreso: 2020-10-12</p>
<p>Medicamento: </p> <p>CLARITROMICINA</p> <p>Concentracion: 50ml</p> <p>Tipo de Medicamento: JARABE</p> <p>Total Unidades Ingreso: 30</p> <p>Unidades Saldo: 30</p> <p>Fecha Ingreso: 2020-10-12</p>

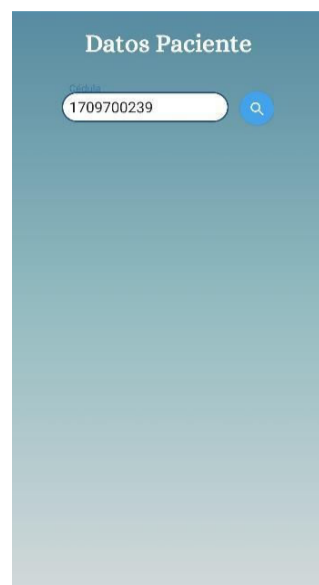
(c)

Figura 3.44. Lista de Medicamentos en Stock Actualizada

3.2.6 PRUEBAS DE VALIDACIÓN DE REQUERIMIENTOS FUNCIONALES: MÓDULO NUEVA ATENCIÓN MÉDICA

3.2.6.1 REQUERIMIENTO FUNCIONAL RF16: INGRESAR DATOS DE PACIENTE

El usuario Médico ingresa la cédula de un paciente y busca si ha sido registrado previamente (Figura 3.45).



The screenshot shows a mobile application interface titled "Datos Paciente". At the top, there is a search bar with the text "1709700239" and a magnifying glass icon to its right. The background is a light blue gradient.

Figura 3.45. Ingreso de Cédula de Nuevo Paciente

Al no estar registrado, ingresa los datos del paciente: apellidos, nombres y sexo (Figura 3.46).



The screenshot shows the "Datos Paciente" form with the following fields: "Cédula" (1709700239), "Apellidos" (Jácome Rosero), "Nombres" (Oscar Alexander), and "Sexo" (radio buttons for "Femenino" and "Masculino", with "Masculino" selected). A "Siguiete" button is at the bottom.

Figura 3.46. Ingreso de Datos Personales de Nuevo Paciente

3.2.6.2 REQUERIMIENTO FUNCIONAL RF17: SELECCIONAR MEDICAMENTO DE RECETA

Para agregar un medicamento a la receta (Figura 3.47), el Médico selecciona el tipo de medicamento, selecciona el medicamento e ingresa las unidades (Parte a). El botón “Agregar” añade el medicamento a la receta y visualiza el mensaje “Medicamento agregado a la receta, puede agregar otro” (Parte b).

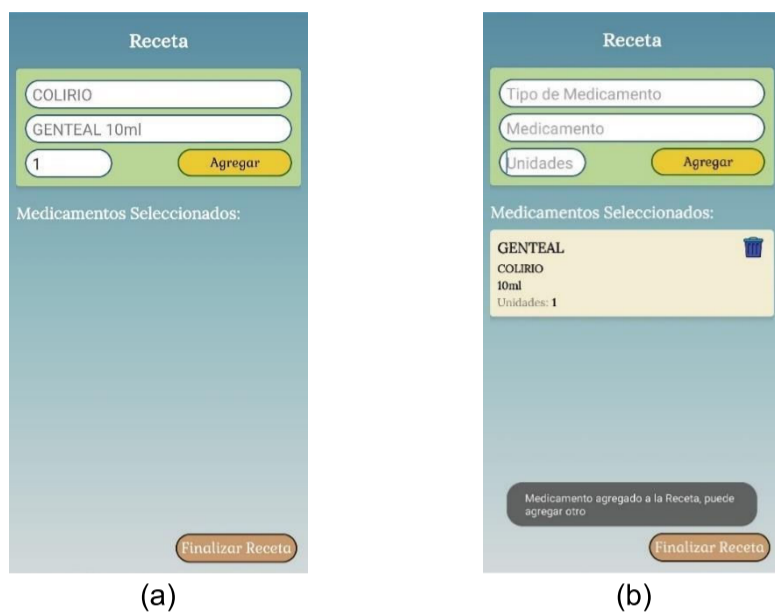


Figura 3.47. Medicamento Agregado a la Receta

Para agregar cuatro medicamentos más a la receta, el médico repite el proceso anterior (Figura 3.48).



Figura 3.48. Cuatro Medicamentos Más Agregados a la Receta

Al repetir de nuevo el proceso (Figura 3.49) para agregar otro medicamento (Parte a) y este no dispone del stock suficiente, dicho medicamento no se agrega a la receta y el médico visualiza el mensaje “Saldo insuficiente de medicamento” (Parte b)



Figura 3.49. Medicamento No Agregado a la Receta por Falta de Stock

3.2.6.3 REQUERIMIENTO FUNCIONAL RF18: CREAR ATENCIÓN MÉDICA

El botón “Finalizar Receta” (Figura 3.49) crea la atención médica en la base de datos, incluyendo al paciente y los medicamentos de receta.

En la base de datos se verifica el registro del paciente en la tabla tbl_Paciente (Figura 3.50).

	id_Paciente	apellidos_Paciente	nombres_Paciente	cedula_Paciente	sexo_Paciente
▶	1	Jácome Rosero	Oscar Alexander	1709700239	Masculino
*	NULL	NULL	NULL	NULL	NULL

Figura 3.50. Paciente Nuevo Registrado en la Base De Datos

En la base de datos se verifica el registro de la atención médica en la tabla tbl_Atencion_Medica (Figura 3.51).

	id_Atencion_Medica	fecha_Atencion_Medica	estado_Atencion_Medica	id_Paciente	id_Medico
▶	1	2020-10-12	Pendiente Entrega	1	1
*	NULL	NULL	NULL	NULL	NULL

Figura 3.51. Atención Médica Nueva Registrada en la Base De Datos

En la base de datos se verifica el registro de los cinco medicamentos de la receta en la tabla tbl_Medicamento_Receta (Figura 3.52), las columnas fecha_Entrega_Medicamento_Receta e id_Dependiente_Farmacia tienen valor NULL, ya que estas columnas se actualizan al registrar la entrega de los medicamentos.

id_Medicamento_Receta	unidades_Medicamento_Receta	fecha_Entrega_Medicamento_Receta	id_Atencion_Medica	id_Medicamento	id_Dependiente_Farmacia
1	1	NULL	1	1	NULL
2	30	NULL	1	5	NULL
3	2	NULL	1	6	NULL
4	4	NULL	1	4	NULL
5	1	NULL	1	7	NULL

Figura 3.52. Medicamentos de Receta Registrados en la Base de Datos

Para crear una segunda atención médica, el médico repite los procesos RF16, RF17 y RF18.

En la base de datos se verifica el registro del segundo paciente en la tabla tbl_Paciente (Figura 3.53).

id_Paciente	apellidos_Paciente	nombres_Paciente	cedula_Paciente	sexo_Paciente
1	Jácome Rosero	Oscar Alexander	1709700239	Masculino
2	Naranjo Ocampo	Angela Marcela	1706453477	Femenino
NULL	NULL	NULL	NULL	NULL

Figura 3.53. Segundo Paciente Registrado en la Base De Datos

En la base de datos se verifica el registro de la segunda atención médica en la tabla tbl_Atencion_Medica (Figura 3.54).

id_Atencion_Medica	fecha_Atencion_Medica	estado_Atencion_Medica	id_Paciente	id_Medico
1	2020-10-12	Pendiente Entrega	1	1
2	2020-10-12	Pendiente Entrega	2	1
NULL	NULL	NULL	NULL	NULL

Figura 3.54. Segunda Atención Médica Registrada en la Base De Datos

En la base de datos se verifica el registro de los cinco medicamentos de la segunda receta en la tabla tbl_Medicamento_Receta (Figura 3.55).

id_Medicamento_Receta	unidades_Medicamento_Receta	fecha_Entrega_Medicamento_Receta	id_Atencion_Medica	id_Medicamento	id_Dependiente_Farmacia
1	1	NULL	1	1	NULL
2	30	NULL	1	5	NULL
3	2	NULL	1	6	NULL
4	4	NULL	1	4	NULL
5	1	NULL	1	7	NULL
6	25	NULL	2	5	NULL
7	30	NULL	2	10	NULL
8	1	NULL	2	8	NULL
9	2	NULL	2	4	NULL
10	2	NULL	2	2	NULL
NULL	NULL	NULL	NULL	NULL	NULL

Figura 3.55. Medicamentos de Segunda Receta Registrados en la Base de Datos

3.2.7 PRUEBAS DE VALIDACIÓN DE REQUERIMIENTOS FUNCIONALES: MÓDULO ATENCIONES MÉDICAS POR ENTREGAR MEDICAMENTOS

3.2.7.1 REQUERIMIENTO FUNCIONAL RF22: BUSCAR PACIENTE REGISTRADO

El usuario Dependiente de Farmacia ingresa la cédula de un paciente y realiza la búsqueda de sus datos registrados (Figura 3.56).

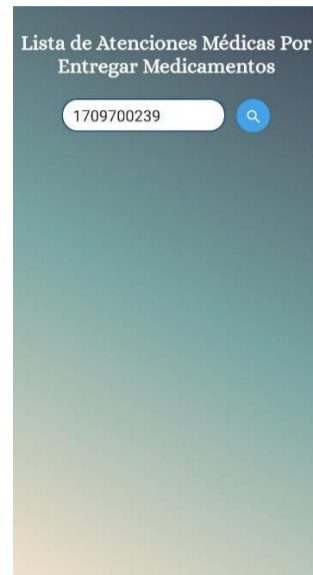


Figura 3.56. Buscar Paciente Registrado por Cédula

El dependiente de farmacia visualiza el nombre del paciente encontrado y la atención médica que tiene pendiente entrega de medicamentos (Figura 3.57).



Figura 3.57. Paciente Registrado y Atención Médica Pendiente

3.2.7.2 REQUERIMIENTO FUNCIONAL RF23: VISUALIZAR ATENCIÓN MÉDICA PENDIENTE DE ENTREGA DE MEDICAMENTOS DE PACIENTE

Al seleccionar la atención médica, se despliegan sus detalles: paciente, médico y medicamentos de receta pendiente de entrega (Figura 3.58).

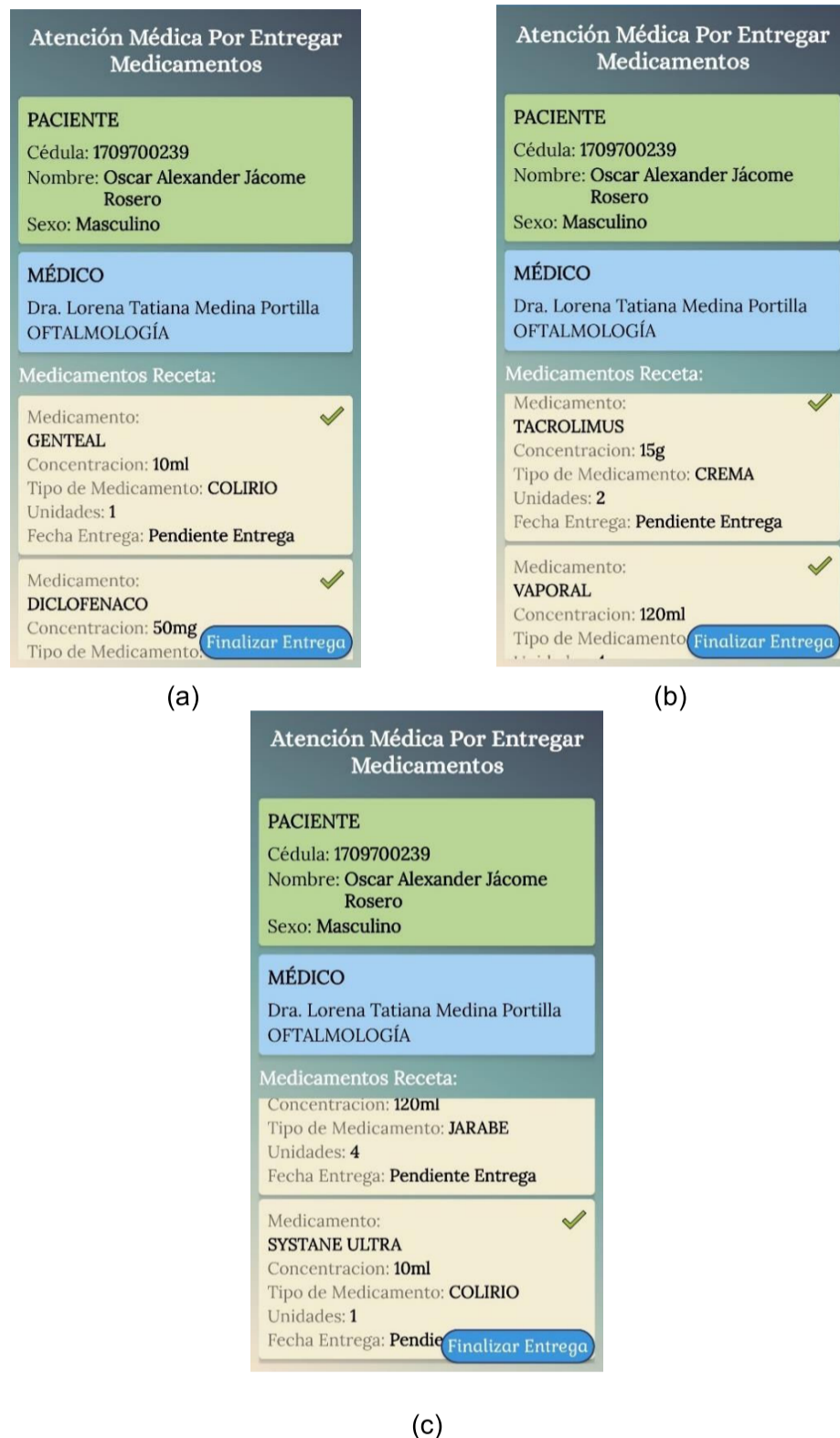


Figura 3.58. Detalles de Atención Médica Pendiente de Entrega de Medicamentos

3.2.7.3 REQUERIMIENTO FUNCIONAL RF24: REGISTRAR ENTREGA DE MEDICAMENTO DE RECETA

Para registrar la entrega (Figura 3.59), el dependiente de farmacia selecciona un medicamento de la receta. Si existe el stock requerido de dicho medicamento, visualiza el mensaje “Si hay disponible la cantidad requerida” y se despliega un cuadro de dialogo de confirmación.

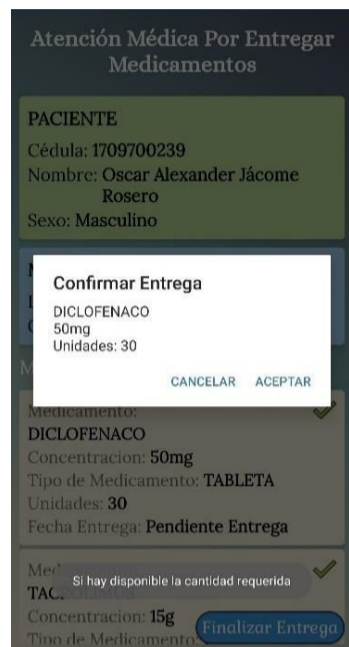


Figura 3.59. Cuadro de Dialogo de Confirmación de Entrega de Medicamento

Al confirmar la entrega, se registra la fecha de entrega del medicamento (Figura 3.60).



Figura 3.60. Registro de Fecha de Entrega de Medicamento de Receta

El proceso de entrega se repite para todos los medicamentos de la receta y en cada uno se visualiza la fecha de entrega (Figura 3.61).

Atención Médica Por Entregar Medicamentos	Atención Médica Por Entregar Medicamentos	Atención Médica Por Entregar Medicamentos
PACIENTE Cédula: 1709700239 Nombre: Oscar Alexander Jácome Rosero Sexo: Masculino	PACIENTE Cédula: 1709700239 Nombre: Oscar Alexander Jácome Rosero Sexo: Masculino	PACIENTE Cédula: 1709700239 Nombre: Oscar Alexander Jácome Rosero Sexo: Masculino
MÉDICO Dra. Lorena Tatiana Medina Portilla OFTALMOLOGÍA	MÉDICO Dra. Lorena Tatiana Medina Portilla OFTALMOLOGÍA	MÉDICO Dra. Lorena Tatiana Medina Portilla OFTALMOLOGÍA
Medicamentos Receta: Medicamento: GENTEAL Concentración: 10ml Tipo de Medicamento: COLIRIO Unidades: 1 Fecha Entrega: 2020-10-12	Medicamentos Receta: Medicamento: DICLOFENACO Concentración: 50mg Tipo de Medicamento: TABLETA Unidades: 30 Fecha Entrega: 2020-10-12	Medicamentos Receta: Medicamento: TACROLIMUS Concentración: 15g Tipo de Medicamento: CREMA Unidades: 2 Fecha Entrega: 2020-10-12
Medicamento: DICLOFENACO Concentración: 50mg Tipo de Medicamento: TABLETA Fecha Entrega: 2020-10-12	Medicamento: TACROLIMUS Concentración: 15g Tipo de Medicamento: CREMA Fecha Entrega: 2020-10-12	Medicamento: VAPORAL Concentración: 120ml Tipo de Medicamento: COLIRIO Fecha Entrega: 2020-10-12

(a)

(b)

(c)

Atención Médica Por Entregar Medicamentos
PACIENTE Cédula: 1709700239 Nombre: Oscar Alexander Jácome Rosero Sexo: Masculino
MÉDICO Dra. Lorena Tatiana Medina Portilla OFTALMOLOGÍA
Medicamentos Receta: Medicamento: VAPORAL Concentración: 120ml Tipo de Medicamento: JARABE Unidades: 4 Fecha Entrega: 2020-10-12
Medicamento: SYSTANE ULTRA Concentración: 10ml Tipo de Medicamento: COLIRIO Fecha Entrega: 2020-10-12

(d)

Atención Médica Por Entregar Medicamentos
PACIENTE Cédula: 1709700239 Nombre: Oscar Alexander Jácome Rosero Sexo: Masculino
MÉDICO Dra. Lorena Tatiana Medina Portilla OFTALMOLOGÍA
Medicamentos Receta: Concentración: 120ml Tipo de Medicamento: JARABE Unidades: 4 Fecha Entrega: 2020-10-12
Medicamento: SYSTANE ULTRA Concentración: 10ml Tipo de Medicamento: COLIRIO Unidades: 1 Fecha Entrega: 2020-10-12

(e)

Figura 3.61. Fecha de Entrega Registrada en Todos los Medicamentos de Receta

3.2.7.4 REQUERIMIENTO FUNCIONAL RF25: ACTUALIZAR ATENCIÓN MÉDICA

Al finalizar el proceso de entrega de todos los medicamentos de la receta, el estado de la atención médica se actualiza y en la parte inferior se visualiza el mensaje “Entregado Completo” (Figura 3.62)



Figura 3.62. Mensaje de Entrega Completa de Medicamentos de Receta

En la base de datos se verifica la actualización del estado de la primera atención médica a “Entregado Completo” (Figura 3.63).

	id_Atencion_Medica	fecha_Atencion_Medica	estado_Atencion_Medica	id_Paciente	id_Medico
▶	1	2020-10-12	Entregado Completo	1	1
	2	2020-10-12	Pendiente Entrega	2	1
•	NULL	NULL	NULL	NULL	NULL

Figura 3.63. Actualización del Estado de la Primera Atención Médica a “Entregado Completo” en la Base de Datos

En la base de datos se verifica la actualización de la fecha de entrega de los cinco medicamentos de receta y el id del Dependiente de Farmacia que registro (Figura 3.64).

	id_Medicamento_Receta	unidades_Medicamento_Receta	fecha_Entrega_Medicamento_Receta	id_Atencion_Medica	id_Medicamento	id_Dependiente_Farmacia
▶	1	1	2020-10-12	1	1	1
	2	30	2020-10-12	1	5	1
	3	2	2020-10-12	1	6	1
	4	4	2020-10-12	1	4	1
	5	1	2020-10-12	1	7	1
	6	25	NULL	2	5	NULL
	7	30	NULL	2	10	NULL
	8	1	NULL	2	8	NULL
	9	2	NULL	2	4	NULL
	10	2	NULL	2	2	NULL

Figura 3.64. Actualización de Entrega Total de Medicamentos de la Primera Atención Médica en la Base de Datos

Para registrar la entrega de medicamentos de la segunda atención médica, el dependiente de farmacia repite los procesos RF22, RF23, RF24 y RF25.

El dependiente de farmacia registró la entrega de tres medicamentos de la receta y en cada uno se verifica la fecha de entrega (Figura 3.65).

The figure shows two screenshots of a mobile application interface for recording medication delivery. Both screenshots are titled "Atención Médica Por Entregar Medicamentos" and share the same patient and doctor information: Patient: Angela Marcela Naranjo Ocampo (ID: 1706453477, Female); Doctor: Dra. Lorena Tatiana Medina Portilla (OFTALMOLOGÍA).

Screenshot (a) shows two medication entries. The first entry is for CLARITROMICINA (50ml concentration, JARABE type, 1 unit, delivered on 2020-10-12). The second entry is for VAPORAL (120ml concentration, partially visible), with a green checkmark next to it. A "Finalizar Entrega" button is visible at the bottom right.

Screenshot (b) shows two medication entries. The first entry is for OMEPRAZOL (40mg concentration, TABLETA type, 30 units, delivered on 2020-10-12). The second entry is for CLARITROMICINA (50ml concentration, partially visible), with a "Finalizar Entrega" button at the bottom right.

(a)

(b)

This screenshot shows the same patient and doctor information as the previous ones. It displays two medication entries. The first entry is for a medication with a concentration of 120ml, JARABE type, 2 units, with a delivery status of "Pendiente Entrega". The second entry is for CANESTEN (20g concentration, CREMA type, 2 units, delivered on 2020-10-12), with a "Finalizar Entrega" button at the bottom right.

(c)

Figura 3.65. Medicamentos Parcialmente Entregados de Segunda Atención Médica

El dependiente de farmacia no registró la entrega de dos medicamentos de la segunda atención médica por falta de stock de los mismos y en cada uno se visualiza el mensaje “No hay disponible la cantidad requerida” (Figura 3.66).

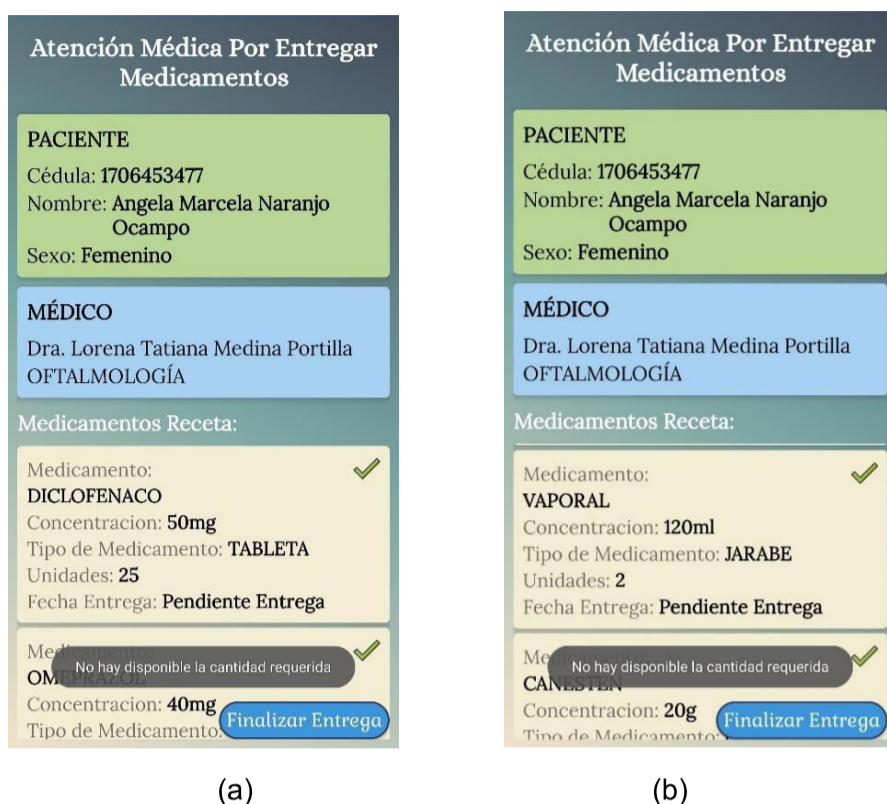


Figura 3.66. Medicamentos No Entregados Por Falta de Stock

En la base de datos se verifica la actualización del estado de la segunda atención médica a “Parcialmente Entregado” (Figura 3.67).

	id_Atencion_Medica	fecha_Atencion_Medica	estado_Atencion_Medica	id_Paciente	id_Medico
▶	1	2020-10-12	Entregado Completo	1	1
	2	2020-10-12	Parcialmente Entregado	2	1
*	NULL	NULL	NULL	NULL	NULL

Figura 3.67. Actualización del Estado de la Segunda Atención Médica a “Parcialmente Entregado” en la Base de Datos

En la base de datos se verifica la actualización de la fecha de entrega de tres de los cinco medicamentos de receta y el id del Dependiente de Farmacia de la segunda atención médica (Figura 3.68).

id_Medicamento_Receta	unidades_Medicamento_Receta	fecha_Entrega_Medicamento_Receta	id_Atencion_Medica	id_Medicamento	id_Dependiente_Farmacia
1	1	2020-10-12	1	1	1
2	30	2020-10-12	1	5	1
3	2	2020-10-12	1	6	1
4	4	2020-10-12	1	4	1
5	1	2020-10-12	1	7	1
6	25	NULL	2	5	NULL
7	30	2020-10-12	2	10	1
8	1	2020-10-12	2	8	1
9	2	NULL	2	4	NULL
10	2	2020-10-12	2	2	1

Figura 3.68. Actualización de Entrega Parcial de Medicamentos de la Segunda Atención Médica en la Base de Datos

3.2.8 PRUEBAS DE VALIDACIÓN DE REQUERIMIENTOS FUNCIONALES: MÓDULO ATENCIONES MÉDICAS REGISTRADAS

3.2.8.1 REQUERIMIENTO FUNCIONAL RF19: VISUALIZAR ATENCIÓN MÉDICA REGISTRADA

El médico visualiza la lista de atenciones médicas que ha registrado, la una tiene estado “Parcialmente Entregado” y la otra “Entregado Completo” (Figura 3.69).



Figura 3.69. Lista de Atenciones Médicas Registradas

Al seleccionar una de las atenciones médicas, se despliegan los detalles de la misma, en este caso la que tiene estado “Parcialmente Entregado”, por lo que se visualizan tres medicamentos con fecha de entrega y los otros dos no (Figura 3.70).

Atención Médica Registrada
Fecha Atención Médica: 2020-10-12

PACIENTE
Cédula: 1706453477
Nombre: Angela Marcela Naranjo Ocampo
Sexo: Femenino

Medicamentos Receta:

Medicamento: **DICLOFENACO**
Concentracion: **50mg**
Tipo de Medicamento: **TABLETA**
Unidades: **25**
Fecha Entrega: **Pendiente Entrega**

Medicamento: **OMEPRAZOL**
Concentracion: **40mg**
Tipo de Medicamento: **TABLETA**
Unidades: **30**
Fecha Entrega: **2020-10-12**

Medicamento:

(a)

Atención Médica Registrada
Fecha Atención Médica: 2020-10-12

PACIENTE
Cédula: 1706453477
Nombre: Angela Marcela Naranjo Ocampo
Sexo: Femenino

Medicamentos Receta:

Medicamento: **CLARITROMICINA**
Concentracion: **50ml**
Tipo de Medicamento: **JARABE**
Unidades: **1**
Fecha Entrega: **2020-10-12**

Medicamento: **VAPORAL**
Concentracion: **120ml**
Tipo de Medicamento: **JARABE**
Unidades: **2**
Fecha Entrega: **Pendiente Entrega**

Medicamento:

(b)

Atención Médica Registrada
Fecha Atención Médica: 2020-10-12

PACIENTE
Cédula: 1706453477
Nombre: Angela Marcela Naranjo Ocampo
Sexo: Femenino

Medicamentos Receta:

Fecha Entrega: **2020-10-12**

Medicamento: **VAPORAL**
Concentracion: **120ml**
Tipo de Medicamento: **JARABE**
Unidades: **2**
Fecha Entrega: **Pendiente Entrega**

Medicamento: **CANESTEN**
Concentracion: **20g**
Tipo de Medicamento: **CREMA**
Unidades: **2**
Fecha Entrega: **2020-10-12**

(c)

Figura 3.70. Detalles de Atención Médica con Estado "Parcialmente Entregado"

3.2.8.2 REQUERIMIENTO FUNCIONAL RF20: REEMPLAZAR MEDICAMENTO DE RECETA POR OTRO

Para reemplazar un medicamento de la receta que no fue entregado por falta de stock por otro equivalente, el médico selecciona dicho medicamento y se despliega un cuadro de dialogo de actualización (Figura 3.71).

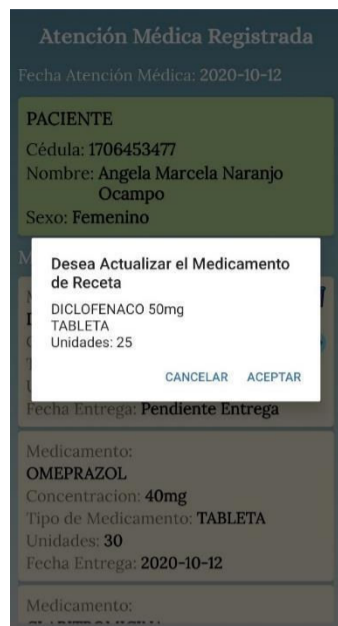


Figura 3.71. Cuadro de Dialogo de Actualización de Medicamento de Receta

A continuación, en la parte inferior el médico ingresa los datos del medicamento equivalente que va a reemplazar al medicamento con insuficiente stock (Figura 3.72).



Figura 3.72. Ingreso de Datos de Medicamento Equivalente

Finalmente, la lista de medicamentos de receta se actualiza, en la que ya consta el medicamento reemplazante y en la parte inferior se visualiza un mensaje “Medicamento de Receta actualizado correctamente” (Figura 3.73).



Figura 3.73. Medicamento de Receta de Actualizado por Equivalente

En la base de datos se verifica la actualización del medicamento de receta de la segunda atención médica en la tabla tbl_Medicamento_Receta (Figura 3.74).

	id_Medicamento_Receta	unidades_Medicamento_Receta	fecha_Entrega_Medicamento_Receta	id_Atencion_Medica	id_Medicamento	id_Dependiente_Farmacia
▶	1	1	2020-10-12	1	1	1
	2	30	2020-10-12	1	5	1
	3	2	2020-10-12	1	6	1
	4	4	2020-10-12	1	4	1
	5	1	2020-10-12	1	7	1
	6	25	NULL	2	9	NULL
	7	30	2020-10-12	2	10	1
	8	1	2020-10-12	2	8	1
	9	2	NULL	2	4	NULL
	10	2	2020-10-12	2	2	1
•	NULL	NULL	NULL	NULL	NULL	NULL

Figura 3.74. Actualización de Medicamento de Receta por Equivalente de Segunda Atención Médica en la Base de Datos

3.2.8.3 REQUERIMIENTO FUNCIONAL RF21: ELIMINAR MEDICAMENTO DE RECETA

Para eliminar un medicamento de la receta que no fue entregado por falta de stock y no existe otro medicamento equivalente para reemplazarlo, el médico selecciona dicho medicamento y se despliega un cuadro de dialogo de eliminación (Figura 3.75).

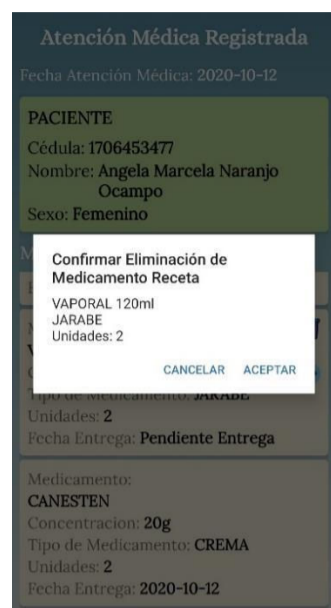


Figura 3.75. Cuadro de Dialogo de Eliminación de Medicamento de Receta

Al confirmar, la lista de medicamentos de receta se actualiza y, el medicamento ha sido eliminado y en la parte inferior se visualiza un mensaje "Medicamento de Receta Eliminado Correctamente" (Figura 3.76).



Figura 3.76. Medicamento de Receta Eliminado

En la base de datos se verifica la eliminación del medicamento de receta de la segunda atención médica en la tabla tbl_Medicamento_Receta (Figura 3.77).

	id_Medicamento_Receta	unidades_Medicamento_Receta	fecha_Entrega_Medicamento_Receta	id_Atencion_Medica	id_Medicamento	id_Dependiente_Farmacia
▶	1	1	2020-10-12	1	1	1
	2	30	2020-10-12	1	5	1
	3	2	2020-10-12	1	6	1
	4	4	2020-10-12	1	4	1
	5	1	2020-10-12	1	7	1
	6	25	NULL	2	9	NULL
	7	30	2020-10-12	2	10	1
	8	1	2020-10-12	2	8	1
	10	2	2020-10-12	2	2	1
*	NULL	NULL	NULL	NULL	NULL	NULL

Figura 3.77. Eliminación de Medicamento de Receta de Segunda Receta en la Base de Datos

Para registrar la entrega del medicamento actualizado por el médico tratante de la segunda atención médica, el dependiente de farmacia repite los procesos RF22, RF23, RF24 y RF25.

El dependiente de farmacia registró la entrega del medicamento que había sido actualizado por el médico, por lo que ya consta con fecha de entrega y en la parte inferior se visualiza el mensaje “Registro de Entrega de Medicamento de Receta” (Figura 3.78).

Atención Médica Por Entregar Medicamentos

PACIENTE
Cédula: 1706453477
Nombre: Angela Marcela Naranjo Ocampo
Sexo: Femenino

MÉDICO
Dra. Lorena Tatiana Medina Portilla
OFTALMOLOGÍA

Medicamentos Receta:

Medicamento:
PARACETAMOL
Concentracion: **80mg**
Tipo de Medicamento: **TABLETA**
Unidades: **25**
Fecha Entrega: **2020-10-12**

Registro de Entrega de Medicamento Receta

Concentracion: **40mg**
Tipo de Medicamento:

Finalizar Entrega

Figura 3.78. Registro de Entrega de Medicamento de Receta Actualizado

En la base de datos se verifica la actualización del estado de la segunda atención médica a “Entregado Completo” (Figura 3.79).

	id_Atencion_Medica	fecha_Atencion_Medica	estado_Atencion_Medica	id_Paciente	id_Medico
▶	1	2020-10-12	Entregado Completo	1	1
	2	2020-10-12	Entregado Completo	2	1
*	NULL	NULL	NULL	NULL	NULL

Figura 3.79. Actualización del Estado de la Segunda Atención Médica a “Entregado Completo” en la Base de Datos

En la base de datos se verifica la actualización de la fecha de entrega de todos medicamentos de receta de la segunda atención médica y el id del Dependiente de Farmacia (Figura 3.80).

	id_Medicamento_Receta	unidades_Medicamento_Receta	fecha_Entrega_Medicamento_Receta	id_Atencion_Medica	id_Medicamento	id_Dependiente_Farmacia
▶	1	1	2020-10-12	1	1	1
	2	30	2020-10-12	1	5	1
	3	2	2020-10-12	1	6	1
	4	4	2020-10-12	1	4	1
	5	1	2020-10-12	1	7	1
	6	25	2020-10-12	2	9	1
	7	30	2020-10-12	2	10	1
	8	1	2020-10-12	2	8	1
	10	2	2020-10-12	2	2	1
*	NULL	NULL	NULL	NULL	NULL	NULL

Figura 3.80. Actualización de Entrega Total de Medicamentos de la Segunda Atención Médica en la Base de Datos

3.2.9 PRUEBAS DE VALIDACIÓN DE REQUERIMIENTOS FUNCIONALES: MÓDULO ATENCIONES MÉDICAS FINALIZADAS

3.2.9.1 REQUERIMIENTO FUNCIONAL RF26: VISUALIZAR ATENCIÓN MÉDICA FINALIZADA

El Dependiente de Farmacia visualiza la lista de atenciones médicas cuyo estado de entrega es “Entregado Completo”, es decir, todos los medicamentos de la receta constan con fecha de entrega registrada (Figura 3.81).

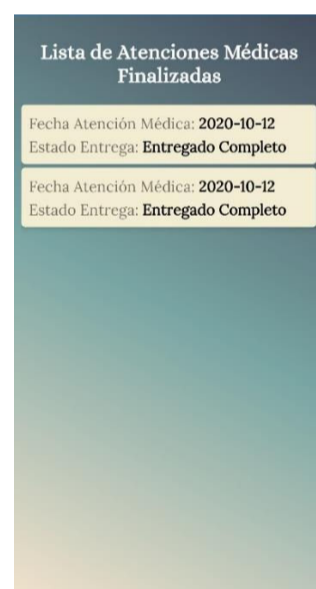


Figura 3.81. Lista de Atenciones Médicas Finalizadas

Al seleccionar una de las atenciones médicas, se despliegan sus detalles: paciente, médico y medicamentos de receta con su respectiva fecha de entrega (Figura 3.82).

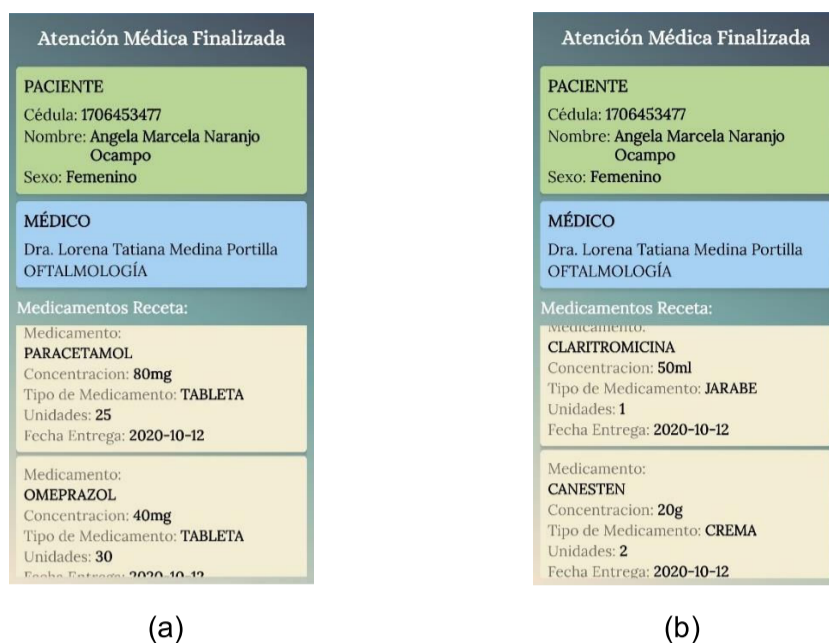


Figura 3.82. Detalles de Atención Médica Finalizada

3.2.10 PRUEBAS DE VALIDACIÓN DE REQUERIMIENTOS FUNCIONALES: MÓDULO HISTORIAL FARMACÉUTICO DE MEDICAMENTO

3.2.10.1 REQUERIMIENTO FUNCIONAL RF12: SELECCIONAR UN MEDICAMENTO PARA HISTORIAL

El usuario Administrador selecciona el tipo de medicamento y el medicamento para el historial (Figura 3.83).

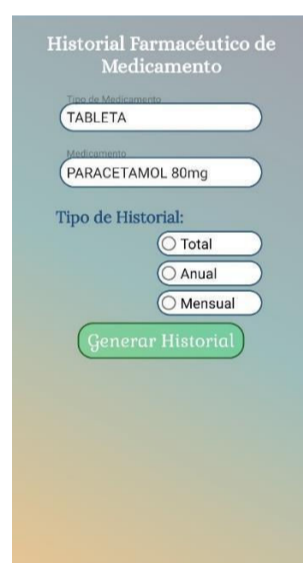


Figura 3.83. Datos de Medicamento Para Historial Farmacéutico

3.2.10.2 REQUERIMIENTO FUNCIONAL RF13: ESCOGER TIPO DE HISTORIAL

El usuario Administrador escoge el tipo de historial entre las tres opciones: total, anual o mensual. Para este ejemplo "Total" (Figura 3.84).

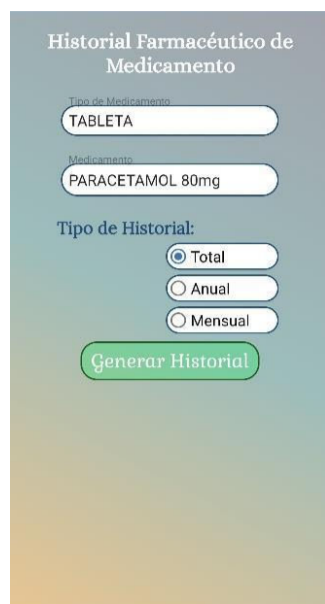


Figura 3.84. Selección de Tipo de Historial: Total

3.2.10.3 REQUERIMIENTO FUNCIONAL RF14: GENERAR HISTORIAL DE MEDICAMENTO

Al generar el historial, el administrador visualiza los datos del medicamento, el tipo de historial, las tablas correspondientes a la entrada y salida del medicamento, y el total de unidades entrante y saliente (Figura 3.85).

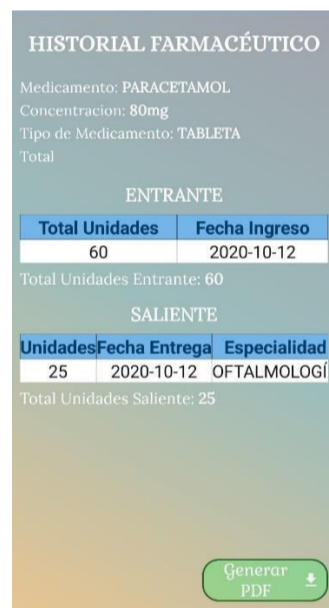


Figura 3.85. Historial Farmacéutico de Medicamento

3.2.10.4 REQUERIMIENTO FUNCIONAL RF15: CREAR PDF CON HISTORIAL DE MEDICAMENTO

Al crear el documento Pdf con la información correspondiente al historial del medicamento, el administrador visualiza una notificación en la parte superior con el nombre del documento PDF y en la parte inferior visualiza el mensaje con el nombre del documento PDF y la ruta de su ubicación (Figura 3.86).

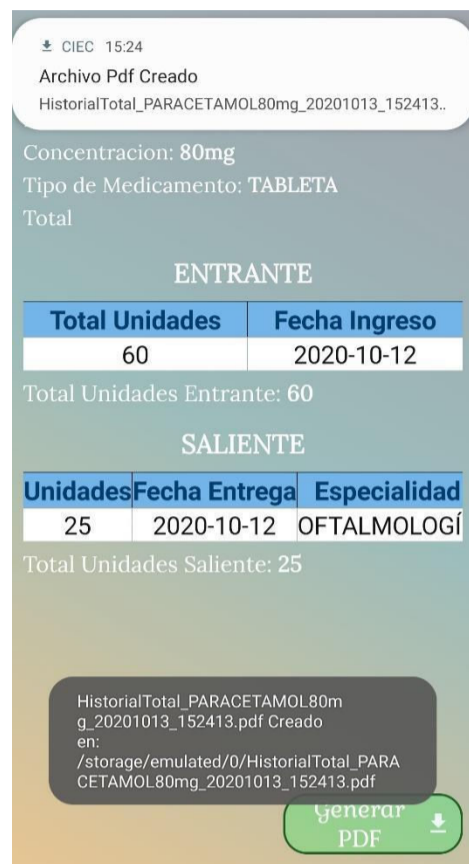


Figura 3.86. Mensaje de Creación de PDF

En el almacenamiento interno del dispositivo, el administrador visualiza en la parte inferior un archivo en Pdf con el nombre del documento creado a través de la aplicación (Figura 3.87).

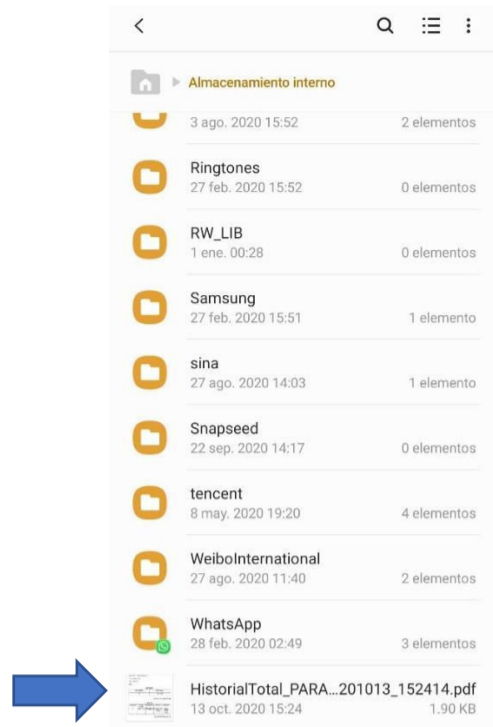


Figura 3.87. Archivo PDF Creado en el Almacenamiento del Dispositivo

El abrir el documento PDF, el administrador verifica que contiene la información correspondiente al historial farmacéutico del medicamento (Figura 3.88).

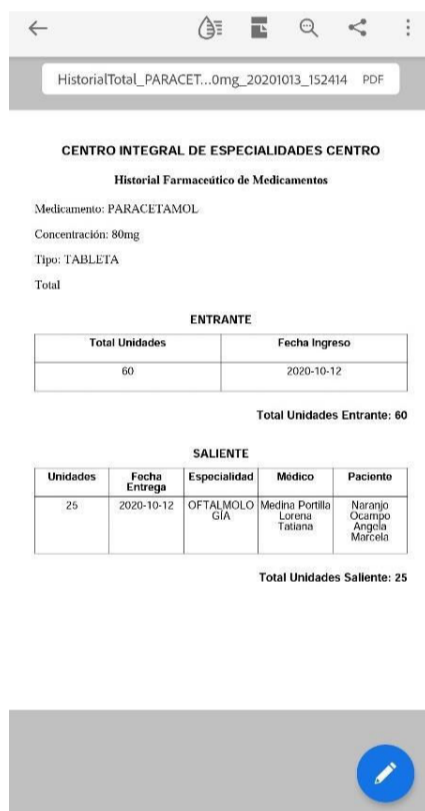


Figura 3.88. Información Contenida en el Archivo PDF

3.3 RESULTADOS DE ENCUESTAS DE VALIDACIÓN DE FUNCIONAMIENTO

Para evaluar el correcto funcionamiento del prototipo se dispuso el aplicativo móvil a doce personas entre administrador, médicos y dependientes de farmacia que laboran en el centro de salud y a cada uno de ellos se les hizo una encuesta luego de utilizar la aplicación. Se establecieron tres tipos de encuestas (ANEXO E) de acuerdo al tipo de usuario: Administrador, Médico y Dependiente de Farmacia. A continuación, se presenta el resultado de dichas encuestas:

Administrador:

Tabla 3.2. Resultado Encuesta de Validación: Administrador (Parte 1 de 2)

Pregunta	Resultado (%)
1. Al realizar el registro de sus datos personales a través de la aplicación móvil, ¿cuál fue el nivel de dificultad?	
a) Muy Fácil	100
b) Fácil	
c) Difícil	
d) Muy Difícil	
2. Usted pudo realizar el inicio de sesión y acceder a la interfaz principal de usuario Administrador	
a) Sí	100
b) No	
c) A veces	
3. La interfaz principal de usuario Administrador disponía de los botones necesarios para acceder a cada módulo	
a) Sí	100
b) No	
c) Parcialmente	
4. En el módulo Nuevo Stock de Medicamento, usted pudo realizar el registro de un nuevo medicamento y su stock	
a) Sí	100
b) No	
5. En el módulo Lista de Medicamentos en Stock, usted pudo visualizar los medicamentos registrados en stock	
a) Sí	100
b) No	
c) Parcialmente	
6. En el módulo Historial Farmacéutico de Medicamento, usted pudo generar el registro farmacéutico de cada medicamento	
a) Sí	100
b) No	
7. En caso de haber elegido la opción a) en la pregunta anterior, usted pudo generar el archivo PDF	
a) Sí	100
b) No	

Tabla 3.3. Resultado Encuesta de Validación: Administrador (Parte 2 de 2)

8. Si se generó el archivo PDF, dicho documento contenía la información respectiva al historial farmacéutico del medicamento especificado en la aplicación móvil	
a) Sí	100
b) No	
9. A su criterio la interfaz gráfica de la aplicación móvil que tan amigable es con el usuario.	
a) Muy Amigable	100
b) Amigable	
c) Poco Amigable	
d) Nada Amigable	
10. Usted considera que las funcionalidades de la aplicación son suficientes para cumplir con los requerimientos planteados	
a) Sí	100
b) No	
c) Parcialmente	

En las Tablas 3.2 y 3.3 se presentan los resultados de la encuesta realizada el Administrador del centro de salud y se pudo verificar que se pudo registrar como usuario de la aplicación sin ningún problema, además, el inicio de sesión fue exitoso. En la parte funcional, el usuario pudo acceder a cada módulo a través de la interfaz principal sin ningún problema y en cada módulo pudo llevar a cabo las tareas determinadas en los requerimientos de forma correcta. A criterio del usuario, la aplicación ofrece una interfaz gráfica muy amigable y cumple con todos los requerimientos, facilitando el uso de la misma.

Médico:

Tabla 3.4. Resultado Encuesta de Validación: Médico (Parte 1 de 2)

Pregunta	Resultado (%)
1. Al realizar el registro de sus datos personales a través de la aplicación móvil, ¿cuál fue el nivel de dificultad?	
a) Muy Fácil	50
b) Fácil	50
c) Difícil	
d) Muy Difícil	
2. Usted pudo realizar el inicio de sesión y acceder a la interfaz principal de usuario Médico	
a) Sí	100
b) No	
c) A veces	

Tabla 3.5. Resultado Encuesta de Validación: Médico (Parte 2 de 2)

3. La interfaz principal de usuario Médico disponía de los botones necesarios para acceder a cada módulo	
a) Sí	100
b) No	
c) Parcialmente	
4. En el módulo Nueva Atención Médica, usted pudo registrar los datos del paciente	
a) Sí	100
b) No	
c) Parcialmente	
5. En caso de haber elegido la opción a) en la pregunta anterior, usted pudo agregar cada medicamento requerido a la receta	
a) Sí	100
b) No	
c) Parcialmente	
6. En el módulo Atenciones Médicas Registradas, usted pudo visualizar las atenciones médicas registradas con anterioridad	
a) Sí	100
b) No	
c) Parcialmente	
7. En caso de haber elegido la opción a) en la pregunta anterior, usted pudo acceder a los detalles de cada atención médica	
a) Sí	100
b) No	
c) Parcialmente	
8. En caso de haber elegido la opción a) en la pregunta anterior, usted pudo visualizar la lista de medicamentos de la receta	
a) Sí	100
b) No	
c) Parcialmente	
9. A su criterio la interfaz gráfica de la aplicación móvil que tan amigable es con el usuario	
a) Muy Amigable	75
b) Amigable	25
c) Poco Amigable	
d) Nada Amigable	
10. Usted considera que las funcionalidades de la aplicación son suficiente para cumplir con los requerimientos planteados	
a) Sí	75
b) No	
c) Parcialmente	25

En las Tablas 3.4 y 3.5 se presentan los resultados de la encuesta realizada a ocho médicos del centro de salud y se pudo verificar que pudieron realizar su registro sin dificultad, ya que la mitad respondió "Muy Fácil" y la otra mitad "Fácil", además, todos los usuarios pudieron realizar el inicio de sesión de manera exitosa. En la parte

funcional, los usuarios accedieron a cada módulo a través de la interfaz principal sin ningún problema y en cada módulo pudieron llevar a cabo las tareas determinadas en los requerimientos de forma correcta. En su gran mayoría los usuarios determinaron que la interfaz gráfica de la aplicación es muy amigable y facilita su uso, siete usuarios estuvieron de acuerdo con la funcionalidad propuesta y uno indicó que parcialmente las cumple.

Dependiente de Farmacia:

Tabla 3.6. Resultado Encuesta de Validación: Dependiente de Farmacia (Parte 1 de 2)

Pregunta	Resultado (%)
1. Al realizar el registro de sus datos personales a través de la aplicación móvil, ¿cuál fue el nivel de dificultad?	
a) Muy Fácil	66,67
b) Fácil	33,33
c) Difícil	
d) Muy Difícil	
2. Usted pudo realizar el inicio de sesión y acceder a la interfaz principal de usuario Dependiente de Farmacia	
a) Sí	100
b) No	
c) A veces	
3. La interfaz principal de usuario Dependiente de Farmacia disponía de los botones necesarios para acceder a cada módulo	
a) Sí	100
b) No	
c) Parcialmente	
4. En el módulo Atenciones Médicas Por Entregar Medicamentos, al ingresar la cédula de un paciente pudo visualizar sus datos y las atenciones médicas recibidas por dicho paciente	
a) Sí	100
b) No	
5. En caso de haber elegido la opción a) en la pregunta anterior, usted pudo acceder a los detalles de cada atención médica, es decir, datos de médico y los medicamentos de la receta	
a) Sí	100
b) No	
c) Parcialmente	
6. En caso de haber elegido la opción a) en la pregunta anterior, usted pudo registrar la entrega de cada medicamento de la receta	
a) Sí	100
b) No	
c) Parcialmente	

Tabla 3.7. Resultado Encuesta de Validación: Dependiente de Farmacia (Parte 2 de 2)

7. En el módulo Atenciones Médicas Finalizadas, usted pudo visualizar las atenciones médicas finalizadas	
a) Sí	100
b) No	
c) Parcialmente	
8. En caso de haber elegido la opción a) en la pregunta anterior, al acceder a los detalles de cada atención médica pudo constatar que cada medicamento constaba con la fecha de entrega	
a) Sí	100
b) No	
c) Parcialmente	
9. A su criterio la interfaz gráfica de la aplicación móvil que tan amigable es con el usuario	
a) Muy Amigable	66,67
b) Amigable	33,33
c) Poco Amigable	
d) Nada Amigable	
10. Usted considera que las funcionalidades de la aplicación son suficientes para cumplir con los requerimientos planteados	
a) Sí	100
b) No	
c) Parcialmente	

En las Tablas 3.6 y 3.7 se presentan los resultados de la encuesta realizada a tres dependientes de farmacia del centro de salud y se pudo verificar que pudieron realizar su registro sin dificultad, ya que dos respondieron “Muy Fácil” y uno respondió “Fácil”, además, todos los usuarios pudieron realizar el inicio de sesión de manera exitosa. En la parte funcional, los usuarios accedieron a cada módulo a través de la interfaz principal de forma correcta y en cada módulo pudieron llevar a cabo las tareas determinadas en los requerimientos con éxito. En su gran mayoría los usuarios determinaron que la interfaz gráfica de la aplicación es muy amigable y los tres estuvieron de acuerdo con la funcionalidad propuesta.

3.4 TABLERO KANBAN FINAL

En la Tabla 3.8 se presenta el tablero Kanban en el cual se especifica que las columnas de Tares Pendientes y Tareas en Proceso se encuentran vacías, es decir, ya no hay tareas por realizarse o realizándose, y la columna de Tareas Realizadas contiene todas las tareas definidas en el desarrollo del proyecto, lo que indica que se ha cumplido con cada una de ellas.

Tabla 3.8. Tablero Kanban Final

Tareas Pendientes	Tareas en Proceso	Tareas Realizadas
		Definición de conceptos teóricos
		Entrevistas a administrador, personal médico y dependiente de farmacia del centro de salud
		Identificación de Requerimientos Funcionales
		Identificación de Requerimientos No Funcionales
		Definición de módulos de la aplicación móvil
		Diseño de Capa de Datos
		Diseño de Capa de Negocio
		Diseño de Capa de Presentación
		Creación de una instancia de EC2 con Windows Server
		Creación de una instancia de RDS con MySql
		Instalación de Node.js y Visual Studio Code en Windows Server
		Instalación de Android Studio
		Codificación de Capa de Datos
		Codificación de Capa de Negocio
		Codificación de Capa de Presentación
		Pruebas de funcionamiento del prototipo
		Encuestas de validación de funcionamiento a usuarios

4 CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- La metodología empleada para el desarrollo de este proyecto fue Kanban, para ello se empleó un tablero Kanban. Permitiendo organizar las actividades que se requieren para el desarrollo de la base de datos, el servicio web REST y la aplicación móvil. El tablero Kanban se actualizó en cada etapa, lo que posibilitó el cumplimiento de todos los requerimientos del prototipo.
- Las consultas SQL establecidas en el servicio web permitieron realizar los procesos de creación, obtención, actualización y eliminación de información en la base de datos y de esta forma a través de Retrofit (cliente Http), los usuarios desde la aplicación accedieron a la información de la base de datos.
- La utilización de Cloud Computing eliminó la necesidad de adquirir e instalar equipos informáticos físicos, permitiendo que la administración y gestión del servidor de la base de datos y el servidor del servicio web se pudieran realizar de manera ágil y desde cualquier sitio.
- La aplicación móvil fue diseñada e implementada en módulos permitiendo cumplir con todos los requerimientos. Para ello, se determinaron las funcionalidades del prototipo identificando y diferenciando las actividades a realizarse en general por todos los usuarios y las específicas de cada tipo de usuario: administrador, médico y dependiente de farmacia.
- Para que una persona se registre como un usuario de la aplicación móvil, entre los datos necesarios consta la Cédula de Identidad, cuya finalidad es que una persona se registre una sola vez con un tipo de usuario y acceda a las funcionalidades específicas para ese tipo de usuario. Evitando acceder a funcionalidades que no le corresponde registrándose con otros tipos de usuario.
- La aplicación permitió a cada usuario registrar, buscar, actualizar, visualizar y eliminar información de forma ágil y rápida, es decir, crear stocks de medicamentos. En base a ello, registrar atenciones médicas incluyendo los datos del paciente y los medicamentos de receta, registrar la entrega de dichos medicamentos de receta. Y finalmente, generar el historial con la información sobre el stock de cada medicamento del centro de salud.
- El documento generado desde la aplicación es muy útil porque permite visualizar la información correspondiente a las tablas de entrada y salida de un medicamento específico y puede ser empleado para conocer si la disposición del medicamento es escasa o no.

- La utilización del cliente Retrofit en la aplicación móvil permitió serializar los datos en Java para que las peticiones y respuestas sean en formato JSON. Permitiendo consumir el servicio web REST y obtener los datos requeridos por el usuario de la aplicación.
- Para el cumplimiento de los requerimientos del prototipo, se implementó la arquitectura en base a tres capas: capa de presentación, capa de negocios y capa de datos. De esta forma los componentes: base de datos, servicio web REST y aplicación móvil se encuentran ubicados en la capa correspondiente, lo que permitió realizar la programación de forma ordenada.
- La programación de la aplicación móvil en Android Studio permitió incluir las librerías necesarias para el diseño de los componentes de las interfaces gráficas, la creación del documento con el historial farmacéutico y la creación del cliente HTTP para consumir el servicio web.
- En el AndroidManifest de Android Studio se establecieron los permisos necesarios para el acceso a Internet y el almacenamiento en la memoria interna del dispositivo. Lo que permitió que el usuario al usar la aplicación móvil pueda acceder a Internet para obtener los datos requeridos y al momento de crear el documento con el historial farmacéutico, este sea almacenado en la memoria interna del dispositivo.
- La elaboración de los mockups permitió que el diseño y la estructuración de los componentes: editores de texto, botones, lista de datos, etc., de cada interfaz gráfica de la aplicación móvil sea realizada de acuerdo a la funcionalidad requerida. Facilitando la implementación de las interfaces gráficas en Android Studio.
- Las pruebas de funcionamiento del prototipo permitieron verificar el cumplimiento de todos los requerimientos establecidos, ya que los tres tipos de usuario: administrador, médico y dependiente de farmacia, pudieron acceder a las funcionalidades correspondientes para cada uno dentro de la aplicación móvil.
- Los resultados de las encuestas realizadas a los usuarios después de utilizar la aplicación móvil, determinaron un alto nivel de aceptación de la misma, ya que es amigable con el usuario y cumple con las funcionalidades correspondientes.

4.2 RECOMENDACIONES

- El Cloud Computing de AWS dispone de una gran cantidad de servicios para el usuario, por lo que es necesario realizar un análisis previo de los requerimientos para evitar la utilización de servicios innecesarios, que pueden recaer en costos adicionales.

- Para la implementación de la aplicación móvil se sugiere identificar de forma única cada componente de las interfaces gráficas, caso contrario, puede producir errores en la codificación de la parte lógica y presentar fallos durante el proceso de compilación del aplicativo móvil.
- Se recomienda comprobar de forma local la correcta conexión entre la base de datos y el servicio web, antes de migrar a los servicios en la nube, caso contrario puede generar resultados erróneos a los usuarios finales que están utilizando la aplicación móvil.
- Para aumentar la escalabilidad de la aplicación, en una nueva versión a futuro se puede añadir un nuevo tipo de usuario: Paciente, añadiendo nuevas funcionalidades para dicho usuario, como visualización de receta médica, agendar próxima atención médica, consultar su historial médico, etc.
- El prototipo se puede utilizar como punto de partida para una versión en el sistema operativo iOS, permitiendo incrementar el número de usuarios de la misma.
- Para futuras versiones se podría implementar el servicio Firebase de la nube de Google, que permite emplear bases de datos en tiempo real que generan actualizaciones automáticamente con los datos más recientes.

5 REFERENCIAS BIBLIOGRÁFICAS

- [1] Ramakrishnan R.; Gehrke J., «Database Management Systems», Third Edition, McGraw-Hill Education, 2003.
- [2] Universidad de Murcia, «Capítulo 9: Sistemas de Gestión de Bases de datos y SIG», [En Línea]. Disponible: https://www.um.es/geograf/sigmur/sigpdf/temario_9.pdf, [Citado: 15-Abr-2019].
- [3] Silberschatz A.; Korth H. F.; Sudarshan S., «Database System Concepts», Sixth Edition, McGraw-Hill, 2010.
- [4] Guru99, «What is DBMS? Application, Types, Example, Advantages, Disadvantages», [En Línea]. Disponible: <https://www.guru99.com/dbms-architecture.html>, [Citado: 16-Abr-2019].
- [5] Tutorials Point, «SQL - Primary Key», [En Línea]. Disponible: <https://www.tutorialspoint.com/sql/sql-primary-key.htm>, [Citado: 16-Abr-2019].
- [6] W3Schools, «SQL FOREIGN KEY Constraint», [En Línea]. Disponible: https://www.w3schools.com/sql/sql_foreignkey.asp, [Citado: 17-Abr-2019].
- [7] TechnologyAdvice, «What is SQL? », [En Línea]. Disponible: <http://www.sqlcourse.com/intro.html>, [Citado: 17-Abr-2019].
- [8] IBM Knowledge Center, «Servicios Web», [En Línea]. Disponible: https://www.ibm.com/support/knowledgecenter/es/SS7K4U_9.0.5/com.ibm.websphere.zseries.doc/ae/cwbs_wbs2.html, [Citado: 22-Abr-2019].
- [9] Kalin M., «Java Web Services: Up and Running», O'Reilly, 2009.
- [10] BBVA, «API REST: qué es y cuáles son sus ventajas en el desarrollo de proyectos», 2016, [En Línea]. Disponible: <https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos>, [Citado: 22-Abr-2019].
- [11] Guru99, «RESTful Web Services Tutorial with Example», [En Línea]. Disponible: <https://www.guru99.com/restful-web-services.html>, [Citado: 23-Abr-2019].
- [12] Bojinov V., «RESTful Web API Design with Node.js», Mumbai, 2015.
- [13] AWS, «What is cloud computing», [En Línea]. Disponible: <https://aws.amazon.com/what-is-cloud-computing>, [Citado: 06-May-2019].
- [14] IBM, «Cloud computing: Guía completa», [En Línea]. Disponible: <https://www.ibm.com/es-es/cloud/learn/cloud-computing>, [Citado: 06-May-2019].
- [15] AWS, «Cloud computing with AWS», [En Línea]. Disponible: <https://aws.amazon.com/what-is-aws>, [Citado: 08-May-2019].

- [16] AWS, «What is Amazon EC2», [En Línea]. Disponible: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html>, [Citado: 08-May-2019].
- [17] AWS, «AWS Free Tier» [En Línea]. Disponible: <https://aws.amazon.com/free/?all-free-tier.sort-by=item.additionalFields.SortRank&all-free-tier.sort-order=asc>, [Citado: 12-May-2019].
- [18] AWS, «What Is Amazon Relational Database Service (Amazon RDS) » [En Línea]. Disponible: <https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Welcome.html>, [Citado: 12-May-2019].
- [19] Techopedia, «Mobile Application (Mobile App)», [En Línea]. Disponible: <https://www.techopedia.com/definition/2953/mobile-application-mobile-app>, [Citado: 22-May-2019].
- [20] Cuello J.; Vittone J., «Diseñando apps para móviles», Cap. 1.
- [21] ThinkMobiles Team, «What are the popular types and categories of apps», [En Línea]. Disponible: <https://thinkmobiles.com/blog/popular-types-of-apps>, [Citado: 23-May-2019].
- [22] DiMarzio J. F., «Beginning Android Programming with Android Studio», Fourth Edition, 2017.
- [23] Aliferi C., «Android Programming Cookbook», 2016.
- [24] Horton H., «Android Programming for Beginners», Packt Publishing, 2015.
- [25] Gerber A.; Craig C., «Learn Android Studio».
- [26] Universidad Carlos III de Madrid, «Software de Comunicaciones: Programación en Dispositivos Móviles Portables - Arquitectura Android», [En Línea]. Disponible: <https://sites.google.com/site/swcuc3m/home/android/generalidades/2-2-arquitectura-de-android>, [Citado: 29-May-2019].
- [27] Android Developers, «Arquitectura de la Plataforma», [En Línea]. Disponible: <https://developer.android.com/guide/platform?hl=es>, [Citado: 29-May-2019].
- [28] Vogel L.; Scholz S.; Weiser D., «Using Retrofit 2.x as REST client - Tutorial», [En Línea]. Disponible: <https://www.vogella.com/tutorials/Retrofit/article.html>, [Citado: 07-Jun-2019].
- [29] Esepe Studio, «Qué es MySQL», [En Línea]. Disponible: <https://www.espestudio.com/noticias/que-es-mysql>, [Citado: 20-Jun-2019].
- [30] MySQL, «MySQL Workbench», [En Línea]. Disponible: <https://www.mysql.com/products/workbench>, [Citado: 20-Jun-2019].
- [31] Visual Studio Code, «Getting Started», Microsoft, [En Línea]. Disponible: <https://code.visualstudio.com/docs>, [Citado: 05-Jul-2019].

- [32] Visual Studio Code, «Node.js tutorial in Visual Studio Code», Microsoft, [En Línea]. Disponible: <https://code.visualstudio.com/docs/nodejs/nodejs-tutorial>, [Citado: 05-Jul-2019].
- [33] Desarrollo Web, «NodeJS», [En Línea]. Disponible: <https://desarrolloweb.com/home/nodejs>, [Citado: 06-Jul-2019].
- [34] Node, «Introduction to Node.js», [En Línea]. Disponible: <https://nodejs.dev>, [Citado: 06-Jul-2019].
- [35] Express, «Integración de la base de datos», [En Línea]. Disponible: <https://expressjs.com/es/guide/database-integration.html>, [Citado: 09-Jul-2019].
- [36] Makhijani C., «Getting Started with Morgan: the Node.js Logger Middleware», 2019, [En Línea]. Disponible: <https://alligator.io/nodejs/getting-started-morgan>, [Citado: 09-Jul-2019].
- [37] Techopedia, «Android SDK», 2019, [En Línea]. Disponible: <https://www.techopedia.com/definition/4220/android-sdk>, [Citado: 21-Jul-2019].
- [38] Android Developers, «App Manifest Overview», [En Línea]. Disponible: <https://developer.android.com/guide/topics/manifest/manifest-intro>, [Citado: 21-Jul-2019].
- [39] Android Developers, «Meet Android Studio», [En Línea]. Disponible: <https://developer.android.com/studio/intro>, [Citado: 27-Jul-2019].
- [40] Luján J. D., «Qué es una Activity», 2014, [En Línea]. Disponible: <https://desarrolloweb.com/articulos/android-que-es-una-activity-o-actividad.html>, [Citado: 27-Jul-2019].
- [41] Master en Desarrollo de Aplicaciones Android, «Ciclo de vida de una actividad», Universidad Politécnica de Valencia, [En Línea]. Disponible: <http://www.androidcurso.com/index.php/tutoriales-android/37-unidad-6-multimedia-y-ciclo-de-vida/158-ciclo-de-vida-de-una-actividad>, [Citado: 29-Jul-2019].
- [42] Be Agile My Friend, «Herramientas para usar el Tablero Kanban por software», [En Línea]. Disponible: <https://www.beagilemyfriend.com/kanban-software>, [Citado: 10-Jun-2019].
- [43] Atlassian, «Kanban: How the kanban methodology applies to software development» [En Línea]. Disponible: <https://www.atlassian.com/agile/kanban>, [Citado: 10-Jun-2019].
- [44] Currículos Exploratorios en TIC, «Modelo Relacional», Colombia, [En Línea]. Disponible: <http://contenidos.sucerman.com/nivel2/web1/unidad2/leccion4.html>, [Citado: 16-Jun-2019].
- [45] Diagramas UML, «Diagrama de casos de uso», [En Línea]. Disponible: <https://diagramasuml.com/casos-de-uso>, [Citado: 14-Jun-2019].

- [46] Geeks for Geeks, «Unified Modeling Language (UML) | Activity Diagrams», [En Línea]. Disponible: <https://www.geeksforgeeks.org/unified-modeling-language-uml-activity-diagrams>, [Citado: 16-Jun-2019].
- [47] Cillero M., «Diagrama de Clases», [En Línea]. Disponible: <https://manuel.cillero.es/doc/metrica-3/tecnicas/diagrama-de-clases>, [Citado: 17-Jun-2019].

6 ANEXOS

Los anexos se incluyen en el CD adjunto al presente documento.

ANEXO A. ENTREVISTAS PARA OBTENCIÓN DE REQUERIMIENTOS

ANEXO B. CÓDIGO BASE DE DATOS

ANEXO C. CÓDIGO SERVICIO WEB REST

ANEXO D. CÓDIGO ANDROID STUDIO

ANEXO E. ENCUESTAS DE VALIDACIÓN DE FUNCIONAMIENTO

ORDEN DE EMPASTADO