



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE CUENCA
CARRERA DE INGENIERÍA DE SISTEMAS

DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA DE INFORMACIÓN
ACCESIBLE DE MEDICINA OCUPACIONAL PARA LA EMPRESA PÚBLICA DE
FARMACIAS MUNICIPALES SOLIDARIAS FARMASOL EP

Trabajo de titulación previo a la obtención
del título de Ingeniero de Sistemas

AUTORES: CHRISTIAN ANDRÉS CALLE ARÍZAGA

JOHNNY MICHAEL ESPINOZA HEREDIA

TUTOR: ING. CRISTIAN FERNANDO TIMBI SISALIMA, Msc.

Cuenca - Ecuador

2022

**CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE
TITULACIÓN**

Nosotros, Christian Andrés Calle Arízaga con documento de identificación N° 0105526404 y Johnny Michael Espinoza Heredia con documento de identificación N° 0107012700; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Cuenca, 12 de julio del 2022

Atentamente,



Christian Andrés Calle Arízaga
0105526404



Johnny Michael Espinoza Heredia
0107012700

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Nosotros, Christian Andrés Calle Arízaga con documento de identificación No. 0105526404 y Johnny Michael Espinoza Heredia con documento de identificación No. 0107012700, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto técnico: “Desarrollo e implementación de un sistema de información accesible de medicina ocupacional para la Empresa Pública de Farmacias Municipales Solidarias FARMASOL EP”, el cual ha sido desarrollado para optar por el título de: Ingeniero de Sistemas, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, 12 de julio del 2022

Atentamente,



Christian Andrés Calle Arízaga
0105526404



Johnny Michael Espinoza Heredia
0107012700

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Cristian Fernando Timbi Sisalima con documento de identificación N° 0103709911, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA DE INFORMACIÓN ACCESIBLE DE MEDICINA OCUPACIONAL PARA LA EMPRESA PÚBLICA DE FARMACIAS MUNICIPALES SOLIDARIAS FARMASOL EP, realizado por Christian Andrés Calle Arízaga con documento de identificación N° 0105526404 y por Johnny Michael Espinoza Heredia con documento de identificación N° 0107012700, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Cuenca, 12 de julio del 2022

Atentamente,



Ing. Cristian Fernando Timbi Sisalima, Msc.
0103709911

AGRADECIMIENTOS

Agradezco primero a Dios por darme la vida y salud para poder culminar con mi trabajo de titulación y cumplir una meta más en mi vida.

De igual forma agradecer a todos los docentes de la carrera por ser un guía durante toda mi vida universitaria y de manera especial a mi tutor de tesis por todo el apoyo, tiempo y conocimientos brindados para realización de este proyecto.

También agradecer a todos mis familiares, amigos por siempre estar pendientes de mi y ser un apoyo incondicional durante todo este trayecto.

Christian Andrés Calle Arízaga

Agradezco primeramente a Dios por darme salud y así guiarme para poder cumplir un escalón más en la vida profesional.

Quiero agradecer a mis familiares, amigos y en especial a mis padres que siempre estuvieron siempre brindando todo su apoyo incondicional e inculcandome valores, conocimientos y enseñanzas.

También agradezco a los docentes de la U.P.S, por compartirnos conocimientos y darnos un apoyo incondicional en todo el desarrollo de nuestro proyecto.

Johnny Michael Espinoza Heredia

DEDICATORIA

Dedico este proyecto de manera especial a mi madre por ser el pilar fundamental que me sostuvo y me alentó con sus consejos a ser una buena persona y a trazarme la meta de llegar a ser un profesional ya que gracias a ella he cumplido todos mis sueños, a mi padre por estar siempre pendiente de mí y no dejarme solo para que a pesar de las adversidades pueda cumplir con mis objetivos planteados.

A mi hermana y sobrinas que siempre estuvieron allí conmigo que con su amor, cariño y palabras de aliento me motivaban a seguir adelante.

A todos mis demás familiares que directa o indirectamente colaboraron para que culminara con este proyecto, a mi tío Daniel que también fue un apoyo importante para culminar un proceso mas de mi vida.

Christian Andrés Calle Arízaga

Dedico este proyecto de manera muy especial a mis padres por brindarme incondicionalmente su apoyo consejos los mismos que me sirvieron como motivación para continuar y jamás rendirme ante las adversidades que hemos tenido con mi carrera profesinal llegando así a cumplir una de mis mayores metas planteadas en mi vida en cual fue llegar a ser un profesional.

A mi hermano William y mi Tia Sandra, que siempre han estado a mi lado apoyándome para poder culminar esta etapa de vida y así poder dar un paso más en mi formación académica, gracias por siempre guiarme y aconsejarme para jamás rendirme ante las adversidades que hemos tenido.

También dedicar a mis tíos, abuelos que siempre han estado apoyándome en los momentos más adversos gracias.

Johnny Michael Espinoza Heredia

RESUMEN

La empresa pública FARMACIAS MUNICIPALES SOLIDARIAS FARMASOL EP, caracterizada por apoyar a la ciudadanía a mejorar su calidad de vida a través de una atención solidaria e integral, como a su vez un bienestar sustentable a sus colaboradores dentro de la entidad. Entonces, la preocupación de la medicina ocupacional por la salud de los trabajadores aplica un rol importante para el beneficio a un ambiente laboral estable y seguro de los mismos. Promoviendo así la protección de la salud de sus colaboradores activos dentro de la empresa, pretendiendo intervenir a tiempo enfermedades graves o mortales que se puedan dar dentro de su desempeño laboral, disminuyendo así condiciones de riesgo de un trabajador. Por tal motivo se ve la necesidad de desarrollar e implementar un SISTEMA MEDICO OCUPACIONAL, enfocado a la atención de sus trabajadores dentro de la empresa el mismo que permitirá mantener un registro de sus consultas referente a su salud dentro de su actividad laboral, con lo cual se realizó el levantamiento de los requerimientos conjuntamente con el medico ocupacional, brindando una mayor facilidad para el desarrollo del sistema, eligiendo como herramientas de desarrollo el framework Angular como front-end debido a su utilidad y versatilidad para el desarrollo de aplicaciones web y una agradable experiencia de usuario y lenguaje de programación java como back-end en su plataforma java JEE permitiendo utilizar una arquitectura n-capas brindando una mayor robustez en el desarrollo del proyecto. Además, se implementó la tecnología WAI-ARIA (Web Accessibility Initiative - Accessible Rich Internet Applications) con el objetivo de obtener una aplicación web accesible, que favorezca la inclusión del usuario final.

Palabras Clave: WAI-ARIA, Medicina ocupacional, Accesibilidad, Framework, Arquitectura n-capas, Salud de los trabajadores, Angular, JAVA EE.

ABSTRACT

The public company MUNICIPAL PHARMACIES IN SOLIDARITY FARMASOL EP, characterized by supporting citizens to improve their quality of life through a supportive and comprehensive care, as well as a sustainable welfare to its employees within the entity. Therefore, the concern of occupational medicine for the health of workers plays an important role for the benefit of a stable and safe working environment. Thus, promoting the protection of the health of its active collaborators within the company, aiming to intervene in time serious or fatal diseases that may occur in their work performance, thus reducing the risk conditions of a worker. For this reason it is necessary to develop and implement an OCCUPATIONAL MEDICAL SYSTEM, focused on the care of its workers within the company, which will allow to keep a record of their consultations regarding their health within their work activity, with which the survey of the requirements was carried out together with the occupational physician, providing greater ease for the development of the system, choosing as development tools the Angular framework as front-end due to its usefulness and versatility for the development of web applications and a pleasant user experience and java programming language as back-end in its java JEE platform allowing the use of an n-tier architecture providing greater robustness in the development of the project. In addition, the WAI-ARIA technology (Web Accessibility Initiative - Accessible Rich Internet Applications) was implemented with the objective of obtaining an accessible web application that favors the inclusion of the end user.

Keywords: WAI-ARIA, Occupational Medicine, Accessibility, Framework, Architecture n-layers, Workers' Health, Angular, JAVA EE.

Contenido

I. MARCO REFERENCIAL	13
1.1. Introducción	13
1.2. Antecedentes	15
1.3. Justificación	15
1.4. Objetivos	16
1.4.1. General	16
1.4.2. Específicos	16
II. MARCO TEÓRICO	18
2.1. Medicina Ocupacional	18
2.1.1. Definición	18
2.1.2. BENEFICIOS	18
2.2. ACCESIBILIDAD WEB	20
2.2.1. Definición	20
2.2.2. NORMAS Y PAUTAS DE ACCESIBILIDAD	20
2.2.3. WCAG 2.0 Capas de orientación	21
2.2.4. Documentos de apoyo de WCAG 2.0	23
2.2.5. Términos importantes en WCAG 2.0	23
2.2.6. WAI-ARIA	24
2.3. USUARIOS CON DISCAPACIDAD VISUAL	25
2.3.1. Definición	25
2.3.2. Tipos	26
2.4. ARQUITECTURA EN CAPAS	28
2.4.1. Introducción	28
2.4.2. SERVICIOS WEB-API REST	28
2.4.3. SERVIDOR DE APLICACIONES	28
2.4.4. JEE	28
2.4.5. ANGULAR	29

2.4.6. POSTGRESQL	30
2.5. LECTORES Y HERRAMIENTAS DE EVALUACIÓN	30
2.5.1. Lectores de Pantalla	30
2.5.2. Herramientas de Evaluación	31
III. MARCO METODOLÓGICO	32
3.1. División de la capa de negocios	34
3.1.1. Entidades de negocio	34
3.1.2. Objeto de acceso a datos	34
3.1.3. Objeto de negocio	35
3.1.4. Fachadas de negocio	35
3.2. Requerimientos	35
3.2.1. Hardware	35
3.2.2. Software	36
3.3. Requerimientos del Aplicativo	36
3.3.1. Identificación de stakeholders	37
3.3.2. Requerimientos Funcionales	38
3.3.3. Requerimientos No Funcionales	47
3.4. Diagrama para la aplicación Web	48
3.4.1. Diagrama Entidad-Relación	48
3.4.2. Diagrama de Clases	51
3.5. Desarrollo	52
3.5.1. SERVIDOR JAVA	52
3.5.2. Creación de la Base de Datos	66
3.5.3. Etapa de Desarrollo Web	68
3.5.4. Descargar de Node.js y NPM	70
3.5.5. Creación de Proyecto en Angular	72
3.5.6. Implementación de Web Service Api REST	75
3.5.7. Tecnología WAI-ARIA	78
IV. RESULTADOS	83
4.1. Módulos del Sistema Web	83
4.1.1. LOGIN	84
4.1.2. MÓDULO USUARIOS DEL SISTEMA	87
4.1.3. MÓDULO MENÚ	88
4.1.4. MÓDULO PACIENTES	89

4.1.5. MÓDULO CONSULTA MEDICA	91
4.1.6. MÓDULO CONSULTA INICIO/PREOCUPACIONAL	94
4.1.7. MÓDULO CONSULTA PERIÓDICA	98
4.1.8. MÓDULO CONSULTA RETIRO/POSOCUPACIONAL	100
4.1.9. MÓDULO CONSULTA REINTEGRO	103
4.1.10. MÓDULO ACCIDENTES	105
4.1.11. HISTORIAL	106
4.1.12. CERTIFICADO DE APTITUD	106
4.1.13. MÓDULO REPORTES	108
4.1.14. UI (User Interface)	114
4.2. Análisis de Resultados	116
V. CONCLUSIONES Y RECOMENDACIONES	118
5.1. Conclusiones	118
VI.REFERENCIAS BIBLIOGRAFICAS	120
6.1. Bibliografía	120
Anexos	123

Índice de tablas

3.1. Requisito Funcional 1	39
3.2. Requisito Funcional 1	40
3.3. Requisito Funcional 2	40
3.4. Requisito Funcional 3	41
3.5. Requisito Funcional 4	41
3.6. Requisito Funcional 5	42
3.7. Requisito Funcional 6	42
3.8. Requisito Funcional 7	43
3.9. Requisito Funcional 8	43
3.10. Requisito Funcional 9	44
3.11. Requisito Funcional 10	44
3.12. Requisito Funcional 11	45
3.13. Requisito Funcional 12	45
3.14. Requisito Funcional 13	46
3.15. Requisito No Funcional 1	47
3.16. Requisito No Funcional 2	47
3.17. Requisito No Funcional 3	47
3.18. Requisito No Funcional 4	48
3.19. Requisito No Funcional 5	48
0.1. Requerimientos del Sistema	127

Índice de figuras

2.1. Percepción a todo color (W3C, 2022a).	27
3.1. Arquitectura para Sistema Médico Ocupacional.	32
3.2. Esquema Lógico de las capas(Sacoto Tapia, 2020).	33
3.3. Diagrama Entidad Relación	50
3.4. Diagrama de clases de Sistema WEB	51
3.5. Diagrama físico.	52
3.6. Esquema Lógico de las capas(Sacoto Tapia, 2020).	53
3.7. Servidor de Aplicaciones	55
3.8. Configuración DataSource	55
3.9. Archivo pom.xml	57
3.10. Configuración de las propiedades del servidor	57
3.11. Método registrar Paciente	59
3.12. Método listar Pacientes	59
3.13. Método registrar Consulta Médica	60
3.14. Método Listar Consultas Médica	60
3.15. Método obtener Consulta Médica	61
3.16. Método Actualizar Consulta Médica	61
3.17. Método Eliminar Consulta Médica	62
3.18. Método Registrar Usuario	63
3.19. Método Listar Usuarios	63
3.20. Método Obtener Usuario	64
3.21. Método Eliminar Usuario	64
3.22. Pantalla inicio PGAdmin	67
3.23. Consola de Administración PGAdmin	67
3.24. Arquitectura en Angula (Gonçalves, 2021).	69
3.25. Instalación de Node js.	70
3.26. Instalación de Node js. cmd	70
3.27. Intalación de Angular Cli	71

3.28. Generando compoente Angular.	74
3.29. Componente Consulta Médica.	75
3.30. Configuración del módulo HttpClient.	76
3.31. Web Service Angular.	77
3.32. Web Service Angular.	77
3.33. Aplicando Aria Role en proyecto.	80
3.34. Tecnología ARIA.	80
3.35. Tecnología ARIA.	81
4.1. Autenticación Usuarios	85
4.2. Menú Principal	85
4.3. Método Login	86
4.4. Método Obtener Usuario por Username	86
4.5. Registrar Usuario	87
4.6. Lista de Usuarios	87
4.7. Menú Principal	88
4.8. Lista de Pacientes	89
4.9. Registrar Paciente	90
4.10. Registrar Consulta Médica	91
4.11. Lista Consultas Médicas	91
4.12. Certificado Médico	92
4.13. Método obtener datos de consulta	93
4.14. PDF MAKE	93
4.15. Registrar Consulta Inicio/Preocupacional	94
4.16. Lista Consultas Inicio/Preocupacional	95
4.17. Registro Exámen Espirometria	96
4.18. Registro Exámen Audiometría	96
4.19. Registro Exámen Laboratorio	96
4.20. Registrar Electrocardiograma	97
4.21. Registrar Radiología	97
4.22. Consulta Inicio/Preocupacional	98
4.23. Registrar Consulta Periódica	98
4.24. Lista Consultas Periódica	99
4.25. Consulta Periódica	99
4.26. Registrar Consulta Retiro/Posocupacional	100
4.27. Lista Consultas Retiro/Posocupacional	101

4.28. Consulta Retiro/Posocupacional	102
4.29. Registrar Consulta Reintegro	103
4.30. Lista Consultas Reintegro	104
4.31. Consulta Reintegro	104
4.32. Registrar Accidente	105
4.33. Lista Accidentes	105
4.34. Accidente	106
4.35. Registrar Certificado Aptitud	107
4.36. Certifico Aptitud	107
4.37. Reporte Módulo Accidentes	108
4.38. Método reporte Accidentes por Fecha	108
4.39. Método reporte Consultas Médica por Fecha	109
4.40. Reporte Módulo Consultas Médica	109
4.41. Método reporte Consultas Inicio-Preocupacional por Fecha	110
4.42. Reporte Módulo Consultas Inicio	110
4.43. Método reporte Consultas Periódica por Fecha	111
4.44. Reporte Módulo Consultas Periódica	111
4.45. Método reporte Consultas Retiro por Fecha	112
4.46. Reporte Módulo Consultas Retiro	112
4.47. Método reporte Consultas Reintegro por Fecha	113
4.48. Reporte Módulo Consultas Reintegro	113
4.49. Barra de Accesibilidad	114
4.50. Ajustar Tamaño del Texto	114
4.51. Componente Estilo de Fuente	115
4.52. Componente Ajustar Contraste	116
0.1. Pruebas con la herramienta WAVE	124
0.2. Pruebas con la herramienta WAVE	125
0.3. Pruebas con la herramienta IBM Equal	125
0.4. Pruebas persona no vidente	126
0.5. Pruebas persona no vidente	126
0.6. Comprobación paquete instalado	128
0.7. Base de Datos Creada	130
0.8. Datasource	136
0.9. Comprobación estado Wildfly	136
0.10. Login	138

0.11. Ventana principal	139
0.12. Lista pacientes	140
0.13. Agregar paciente	140
0.14. Lista de pacientes para consulta	141
0.15. Lista de consultas por paciente	142
0.16. Reportes	143
0.17. Barra de accesibilidad	144

Capítulo I

MARCO REFERENCIAL

1.1. Introducción

Hoy en día las empresas, particularmente FARMACIAS MUNICIPALES SOLIDARIAS FARMASOL EP que tiene un punto muy importante como es la Medicina del trabajo o también conocido hoy en día como la Medicina Ocupacional el mismo que es una rama interdisciplinaria con el objetivo de analizar las condiciones de un trabajador el mismo que está expuesto a innumerables riesgos que pueden afectar directamente la salud. También, en la actualidad tener un manejo adecuado de la información es relativamente importante más aun dentro de una empresa por lo que contar con un sistema que ayuden a gestionar toda la información de cada uno de los pacientes seria transcendental, es por ello que se implementara un software en donde le permita al médico ocupacional realizar consultas médicas, preocupacionales entre otras, obteniendo así grandes ventajas, tanto para la empresa como para el Doctor Ocupacional ya que mediante esta implementación de este software le permitirá optimizar el tiempo de atención a un paciente, como también llevar un control de cada uno

de los pacientes como por ejemplo tener un historial de las consultas que se a realizado un determinado paciente de forma rápida, ordenada y eficiente ya que todo será sistematizado.

(Violeta María García Tijerino, 2014) “Es la promoción y el mantenimiento del más alto grado de bienestar físico, mental y social de todas las/los trabajadores a través de la prevención y el control de los factores de riesgo y de la adaptación del trabajo al hombre. Esto se logra con abordaje multidisciplinario e implicaciones de profesionales con diferentes especialidades.” Es importante dedicarse a la anticipación, reconocimiento, evaluación y control de aquellos factores ambientales o estreses que surgen en el lugar de trabajo, que pueden causar enfermedad, deterioro de la salud o incomodidad e ineficiencia marcada entre los trabajadores.

Promoviendo el cuidado de la salud de sus colaboradores activos dentro de la empresa, pretendiendo intervenir a tiempo lesiones, enfermedades graves o mortales que se puedan dar dentro de su desempeño laboral, disminuyendo así condiciones el riesgo de una persona dentro de sus actividades laborales; El presente documento proporcionara una introducción a los conceptos de cada una de las tecnologías utilizadas e indispensables para el desarrollo del software, de igual forma se mostrara todos los procesos ineludibles que se persiguió para el aplicativo web dentro de nuestro proyecto de titulación.

1.2. Antecedentes

En el ámbito de la medicina, específicamente el área de la medicina ocupacional es indispensable en una empresa, ya que gracias a ello se lleva un control con respecto a la salud de un paciente, surgiendo en la actualidad el interés por parte de las empresas contar con sistemas Médicos en Salud Ocupacional que permitan gestionar y vigilar la salud de los miembros de su organización ayudando a trabajar con toda la información que se maneja y genera al respecto. Por ello con este proyecto se busca ayudar concretamente al médico tanto en su carrera profesional y a la empresa a solventar el problema de manejo de pacientes, con lo que el proyecto pretende ser una herramienta/sistema que permita mantener un control, gestión e información de cada uno de sus trabajadores. También llegando a reducir considerablemente el tiempo en las actividades médicas a realizar, en la cual un médico pueda saber al instante toda la información de un paciente que previamente fue atendido.

1.3. Justificación

Debido al incremento tecnológico se abre la posibilidad de creación de nuevas Tecnologías de Información (TI) que logren facilitar y optimizar el trabajo dentro de una empresa reduciendo tiempos y costos. Por lo tanto, implementar tecnologías o nuevos sistemas ayuda a diversos campos como en este caso a la medicina ocupacional que controla y gestiona el buen estado de salud de los pacientes(trabajadores) a llevar de manera eficiente-automatizada toda esa información dentro de la empresa. Así se pretende implementar un sistema me-

dico accesible eficiente en el área ocupacional que permita al médico automatizar las distintas actividades Medicas logrando preservar/mantener un control y gestión adecuado de la información (accidentes, como también solventar el inconveniente que posee el gestionar los datos en archivos físicos (operaciones de escritura) u hojas de Excel dentro de la empresa que en ocasiones son difíciles de tratar por el gran volumen de información y ralentizan su trabajo, logrando así con el desarrollo de este sistema llevar un adecuado control de las actividades Medico Ocupacionales. enfermedades, exámenes, reportes etc.) de sus pacientes El sistema además considerará el tema de accesibilidad que brinde la posibilidad de ser usado o empleado por el mayor número de personas ya que en la actualidad se vuelve una característica muy importante implementada en diversos tipos de desarrollo de software, permitiendo que personas especialmente con algún tipo de discapacidad visual sean capaces de interactuar con estas plataformas.

1.4. Objetivos

1.4.1. General

Desarrollar e implementar un sistema de información accesible de Medicina Ocupacional para la empresa FARMASOL EP.

1.4.2. Específicos

- Estudiar y conocer los aspectos principales y fundamentos relacionados a la Accesibilidad en Sistemas de Información con énfasis en personas con discapacidad visual.

- Definir y documentar los requerimientos (funcionales y no funcionales) del sistema a desarrollar.
- Diseñar el modelo conceptual (utilizando UML) y la interfaz gráfica a implementarse en el sistema.
- Desarrollar el sistema de acuerdo con los requerimientos y con el cumplimiento de estándares de accesibilidad.
- Elaborar y ejecutar un plan de pruebas que permita validar la funcionalidad del sistema.
- Implementar el sistema en la infraestructura de la Empresa FARMASOL EP y capacitar a los usuarios en el uso de este.

Capítulo II

MARCO TEÓRICO

2.1. Medicina Ocupacional

2.1.1. Definición

"Se refiere a la rama de la medicina que implica mantener la salud y la seguridad de los trabajadores y empleados en el lugar de trabajo. Incluye el tratamiento de lesiones y enfermedades y también su prevención. Por otra parte, el objetivo secundario de la medicina ocupacional es impulsar o mantener la productividad, estos diagnostican, manejan y previenen lesiones, así como enfermedades que han sido causadas o exacerbadas por factores en el lugar de trabajo. Se centran en todos los aspectos de los efectos del trabajo en la salud y también en la salud en el lugar de trabajo."(Tuček, 2018).

2.1.2. BENEFICIOS

Según Burton (2010) indica que entre los beneficios para bienestar y salud del empleo de la medicina ocupacional se pueden describir los siguientes:

- Mejorar la salud de los empleados, conduce a una mejor productividad de la fuerza laboral, un mejor desempeño comercial y en última instancia, una economía sólida.
- La Salud Ocupacional busca promover, mantener tanto el bienestar físico, mental y social de los trabajadores. Explora los efectos del trabajo en la salud de una persona y también los efectos de la salud en su capacidad para trabajar.
- Los profesionales de salud ocupacional brindan asesoramiento independiente a la gerencia y los empleados acerca de prevención de enfermedades y lesiones asociadas con el trabajo, asegurando que los empleados estén en condiciones de desempeñar sus funciones, mejorando los entornos y las culturas laborales, optimizando la gestión y la rehabilitación de los trabajadores afectados.
- Los servicios de salud ocupacional también ayudan a las organizaciones a cumplir con sus deberes legales según la legislación laboral de cada país.

2.2. ACCESIBILIDAD WEB

2.2.1. Definición

Hace referencia a que personas con discapacidad pueden usar la Web como alguien con plena capacidad. Específicamente accesibilidad web significa que personas con discapacidad perciban, comprendan, navegen e interactuen con la web y que puedan contribuir a la web. La accesibilidad web beneficia a otros, incluidas las personas mayores con habilidades cambiantes debido, por ejemplo al envejecimiento.(W3C, 2021).

Cuando se habla de accesibilidad web, en realidad se está refiriendo a accesibilidad digital, ya que los mismos principios se aplican a las aplicaciones móviles y otros productos digitales. En términos simples, la accesibilidad digital es garantizar que todos puedan acceder a su sitio web y aplicaciones móviles, independientemente de las barreras, siempre que el contenido sea accesible en todas las direcciones posibles; es asegurarse de no haber excluido a personas con discapacidades motoras o cognitivas, visuales, auditivas de su contenido. (W3C, 2021).

2.2.2. NORMAS Y PAUTAS DE ACCESIBILIDAD

A medida que la tecnología avanza a un ritmo acelerado, lo digital es el camino a seguir. Muchas empresas ahora están evolucionando y expandiéndose hacia el mundo digital. La gente desea un servicio al que se pueda acceder desde la comodidad de su hogar, en la palma de su mano. Es por ello, "que las Pautas de Accesibilidad al Contenido Web (WCAG) 2.0 cubren una amplia gama de recomendaciones para hacer que el contenido Web sea más accesible. Seguir estas pautas hará que el contenido sea accesible para una gama más amplia de personas con discapacidades, que incluyen ceguera y baja visión, sordera y pérdida auditiva, problemas de aprendizaje, limitaciones cognitivas, movimiento limitado, discapacidades del habla, fotosensibilidad y combinaciones de estos. Seguir estas pautas también hará que el contenido de una web sea más útil para los usuarios en general."(W3C, 2021).

"WCAG 2.0 se desarrolla a través del proceso W3C en cooperación con personas y organizaciones de todo el mundo, con el objetivo de proporcionar un estándar compartido para la accesibilidad del contenido web que satisfaga las necesidades de personas, organizaciones y gobiernos a nivel internacional. WCAG 2.0 se basa en WCAG 1.0 [WCAG10] y está diseñado para aplicarse ampliamente a diferentes tecnologías web ahora y en el futuro y para poder probarse con una combinación de pruebas automatizadas y evaluación humana."(W3C, 2021).

Según W3C (2021) Accesibilidad web no solo depende de un contenido accesible, sino también de otros agentes de usuario y navegadores web accesibles. Las herramientas de creación también tienen un papel importante en la accesibilidad web.

2.2.3. WCAG 2.0 Capas de orientación

Las personas y organizaciones que utilizan las WCAG varían desde diseñadores y desarrolladores web, legisladores, agentes de compras, profesores y estudiantes. Con el fin de satisfacer las diversas necesidades de esta audiencia, se proporcionan varios niveles de orientación incluyendo pautas generales, criterios de éxito comprobables y principios generales como una colección de técnicas suficientes, de asesoramiento y fallas más comunes todo esto documentado con ejemplos, enlaces de recursos y código. (W3C, 2022b).

- **Principios:** La parte superior cuenta o existen cuatro principios base que se proporcionan para la accesibilidad web: operable, comprensible, perceptible y sólido. (W3C, 2022b)
- **Pautas:** "Las 12 pautas brindan los objetivos básicos para que los autores puedan trabajar para hacer que el contenido sea más accesible enfocado principalmente a los usuarios con discapacidades diferentes. Dichas pautas no son comprobables, pero proporcionan tanto el marco y los objetivos generales que ayuden a los autores o creadores a comprender los distintos criterios de éxito e implementar las técnicas de una manera correcta."(W3C, 2022b)
- **Criterios de éxito:** Se brinda o proporciona los criterios de éxito comprobables para permitir el uso de WCAG 2.0 en cada directriz donde requisitos como las distintas pruebas

de conformidad son necesarios, como en la especificación de diseño, la compra, la regulación y los acuerdos contractuales. Para satisfacer los requerimientos de diferentes grupos y diferentes situaciones, se establecen niveles de conformidad, específicamente tres: A (más bajo), AA y AAA (más alto). (W3C, 2022b)

- **Técnicas suficientes y de asesoramiento:** Se ha documentado una gran variedad de técnicas con lo que respecta a cada pauta y criterios de éxito en WCAG 2.0. Estas técnicas de carácter informativo se dividen en dos categorías: las que permiten cumplir de manera suficiente con los criterios de éxito y las técnicas que son de asesoramiento. Las técnicas de asesoramiento van más allá de lo que exigen los criterios de éxito individuales las cuales permiten a los autores comprender o abordar las directrices de mejor manera. Otras técnicas de asesoramiento engloban barreras de accesibilidad que no están incluidas en los criterios de éxito comprobables. Cuando es de conocimiento acerca de fallas comunes, también se documentan. (W3C, 2022b)

Cada una de las capas de orientación (principios, criterios de éxito, pautas, técnicas suficientes y de asesoramiento) trabajan de manera conjunta para brindar asesoramiento sobre cómo hacer para que todo nuestro contenido sea más accesible. Se alienta a los autores a revisar y aplicar el mayor número de capas que puedan, incluidas las distintas técnicas de asesoramiento, para abarcar la gama más amplia posible de usuarios abordando mejor sus necesidades.(W3C, 2022b).

Se debe tener en consideración que aun incluso el contenido que se ajuste al nivel de accesibilidad más alto (AAA) no será accesible con todos los tipos, grados o combinaciones de personas con discapacidad, particularmente en las áreas de aprendizaje y lenguaje cognitivo. Se incentiva a los autores a tener en cuenta la gama completa de técnicas, como las técnicas de asesoramiento así como a buscar asesoramiento relevante para garantizar un contenido web mas accesible, en la medida de lo posible con las mejores prácticas actuales, para esta comunidad. Para ayudar a encontrar el contenido más adecuado a sus necesidades los usuarios pueden hacer uso de los metadatos. (W3C, 2022b).

2.2.4. Documentos de apoyo de WCAG 2.0

Según W3C (2022) documento desarrollado para satisfacer las necesidades de aquellos que necesitan un estándar técnico estable y referenciable. Otros documentos (documentos de apoyo), están basados en el WCAG 2.0 y abordan propósitos importantes, incluida la capacidad de actualizarse para describir cómo se aplicarían las WCAG o implementarían con nuevas tecnologías. Los documentos de apoyo incluyen:

- **Cómo cumplir con las WCAG 2.0:** Existen cuatro principios base que se proporcionan para la accesibilidad web: operable, comprensible, perceptible y sólido. (W3C, 2022b).
- **Entendiendo las WCAG 2.0 :** Existe un documento breve para la comprensión de WCAG 2.0. para cada directriz y criterio de éxito siendo una guía para la implementación y entendimiento de las WCAG 2.0.
- **Técnicas para WCAG 2.0:** Documento con un conjunto de fallas y técnicas comunes independientemente separadas incluyendo descripción, ejemplos, código y pruebas.
- **Los documentos WCAG 2.0:** Diagrama y descripción de la relación o vínculo que tiene los documentos técnicos.

2.2.5. Términos importantes en WCAG 2.0

Se incluyen tres términos indispensables en WCAG 2.0 que son diferentes de WCAG 1.0. Cada uno de estos se detalla brevemente a continuación."(W3C, 2022b).

- **Página web:** .Existen cuatro principios base que se proporcionan para la accesibilidad web: operable, comprensible, perceptible y sólido. Estos cuatro principios sobre accesibilidad deben ser comprendidos."(W3C, 2022b).
- **Determinado programáticamente:** "Gran variedad de criterios de éxito requieren que ya sea ciertos aspectos o el total de el contenido se pueda determinar programáticamente. Esto significa que un contenido pueda entregarse de manera que los agentes de usuario,

incluidas las TA (tecnologías de asistencia), puedan percibir y presentar en diferentes modalidades dicha información a los usuarios. (W3C, 2022b).

- **Accesibilidad admitida:** El uso de una tecnología de una manera que sea compatible con la accesibilidad significa que funciona con tecnologías de asistencia (AT) y a su vez con características de accesibilidad en navegadores, sistemas operativos y otros agentes de usuario. Solo se puede confiar en que las características tecnológicas se ajusten a los criterios de éxito de las WCAG 2.0 compatible con la accesibilidad mediante sus uso. Las funciones tecnológicas se pueden usar de maneras no compatibles con la accesibilidad (no funcionan con tecnologías de asistencia, etc.) siempre que no se dependa de ellas para cumplir con ningún criterio de éxito, es decir, la misma funcionalidad o información también está disponible de alguna otra manera que sea compatible. (W3C, 2022b).

2.2.6. WAI-ARIA

Según WAI-ARIA (2022) “Es el conjunto de aplicaciones que identifica el cómo hacer un implícito web como de igual forma las apps coexistan en lo más sencilla posible para las personas que presten alguna discapacidad, brinda la posibilidad de realizar apps dinámicas donde se tenga un control de interfaz para el usuario final, desarrollando con herramientas como HTML 5 como también JavaScript y entre otras tecnologías coherentes. Sin WAI-ARIA, ciertas funciones utilizadas en las páginas web, prácticamente no estarían útiles para el usuario final que tenga alguna discapacidad.”

Por otra parte, “proporciona un marco para agregar atributos para identificar características para la interacción del usuario, cómo se relacionan entre sí y su estado actual. Adicionalmente, describe técnicas de navegación para marcar regiones y estructuras web comunes como menús, contenido principal, contenido secundario, información de banner y otros tipos de estructuras web. Por ejemplo, con estas aplicaciones los desarrolladores pueden identificar regiones de páginas y permitir que los usuarios del teclado se muevan fácilmente entre regiones, en lugar de tener que presionar Tab muchas veces.” (WAI-ARIA, 2022)

"WAI-ARIA proporciona algunos elementos para llevar un mejor control de cada elemento que contruye esta tecnología "

- Un ROL para poder indentificar un tipo de widget que puede optar.
- Un ROL para identificar una estructura dentro de una pagina web.
- Propiedades para ver los cambios de estado de cada elemento.
- "Trnsformar lo eventos de teclado en acciones que proporcionen una accion dentro un pagina web"

2.3. USUARIOS CON DISCAPACIDAD VISUAL

2.3.1. Definición

Según W3C (2022a) indica que la “Baja visión” se refiere a impedimentos visuales distintos a la ceguera. Otros términos, como "parcialmente videntez "discapacitado visual", se utilizan en diferentes contextos."

Segun (W3C, 2022a) “En muchos contextos, la baja visión solo incluye deficiencias que no se corrigen con anteojos regulares, lentes de contacto, medicamentos o cirugía. Por ejemplo, la incapacidad para enfocar objetos cercanos al espectador, pero que se puede superar con anteojos para leer, no se considera baja visión en algunos contextos de derechos de discapacidad. En estos contextos, la baja visión a menudo se define como una discapacidad visual que interfiere con la capacidad de una persona para realizar las actividades cotidianas.”

En otros contextos, como los beneficios de los programas sociales, las personas se clasifican como personas con baja visión o “legalmente ciegas”. Muchas personas que son legalmente ciegas tienen una visión utilizable y pueden leer un texto cuando se muestra de manera óptima

para ellos.

2.3.2. Tipos

Según W3C (2022a) Existe un sin numero de elementos que pueden afectar a uuna persona que presente este tipo de discapacidad, en las cuales presentamos a continuación.

- **Susceptibilidad a la luz:** Existe personas que soy muy sensibles a la luz oh contraste a un monitos oh ya sea un terminal el miso que puede afectar poder utilizar correctamente un sistema web.
- **Afectividad en el contraste:** Tener la capacidad de distinguir los diferentes colores que puede llegar a tner una página web.
- **Enfoque de color** Algunas personas, tiene a tener un enfoque a colores bastantes fuertes oh ya sea que algunos otros casos que algunos otros colores son muy degradados y es por ello que la tecnologia de wai-aria verifica dicha enfoque oh contraste.
- **Visión de funcionalidad** Según W3C (2022a) "la mayoría de las personas con baja visión tienen múltiples impedimentos visuales. Por ejemplo, las personas con degeneración macular suelen desarrollar pérdida del campo central "puntos ciegos", agudeza visual deficiente, baja sensibilidad al contraste y alta sensibilidad a la luz. Otros factores, como la inflamación, la medicación, la fatiga y el entorno, influyen en la visión efectiva de una persona en una situación determinada."

A veces, las personas tienen control sobre los factores ambientales, por ejemplo, pueden cambiar la iluminación de la habitación, configurar el brillo del dispositivo, moverse a un lugar sombreado o cambiar el ángulo de la pantalla. A veces, las personas tienen poco o ningún control sobre el medio ambiente, por ejemplo, cuando usan un quiosco público (W3C, 2022a).

Las tareas también pueden afectar la visión funcional de una persona. Por ejemplo, una persona puede distinguir letras en un tamaño de punto determinado en una prueba de visión en la que lee solo una letra a la vez, pero en realidad no puede leer bloques de texto

en ese mismo tamaño de punto. Esta es la diferencia entre legibilidad y facilidad de lectura. La legibilidad está relacionada con la percepción del texto mediante la distinción de letras. Facilidad de lectura está relacionado con la lectura y comprensión de información textual. Por lo tanto, el texto podría ser algo legible para una persona, pero no funcionalmente legible: con esfuerzo, el usuario podría distinguir una letra de otra, pero no podría leer oraciones de manera efectiva (W3C, 2022a).

.Algunos problemas de accesibilidad de baja visión están directamente relacionados con la visión funcional y algunos están relacionados con adaptaciones para discapacidades visuales. Por ejemplo, cuando una persona con baja agudeza visual y campo de visión completo, usa el zoom o la ampliación, tiene algunos problemas similares a los de las personas con pérdida de campo periférico, ya que una cantidad considerable de información puede moverse fuera de su campo de visión como resultado de su solución a la agudeza visual reducida."(W3C, 2022a).



Figura 2.1: Percepción a todo color (W3C, 2022a).

2.4. ARQUITECTURA EN CAPAS

2.4.1. Introducción

2.4.2. SERVICIOS WEB-API REST

Según Kurtz y Wortman “es un estilo arquitectónico que define un conjunto de restricciones que se utilizarán para crear servicios web” REST API es una de las sin número de formas que permiten realizar una comunicación con la capa del servidor como la capa de presentación donde a través de sus diferentes métodos como POST GET O PUT pueden ser almacenados o recuperados y vistos a través de una plataforma hacia el usuario final.

Esta libertad y flexibilidad inherentes al diseño de API REST le permiten crear una API que satisfaga las necesidades de los diseñadores y al mismo tiempo, satisfaga las necesidades de clientes. A diferencia de SOAP, REST no está limitado a XML, sino que puede devolver XML, JSON, YAML o cualquier otro formato según lo que solicite el cliente y en contraste con RPC, no se requiere que los usuarios conozcan los nombres de los procedimientos o parámetros específicos en un orden específica (Kurtz y Wortman, 2014).

2.4.3. SERVIDOR DE APLICACIONES

Según García (2019) “Es diseñado específicamente para ejecutar aplicaciones” El mismo que incluye tanto el hardware como el software que proporcionan un entorno para que se ejecuten los programas, este funciona junto con otros elementos, como servidores de medios y controladores de sesión, proporcionando lógica e inteligencia comercial para brindar servicios complementarios a usuarios de clientes empresariales y suscriptores de servicios residenciales.

2.4.4. JEE

Según ORACLE (2022) “Es una compilación de API de Java en la cual tiene como propietario de Oracle.”

2.4.5. ANGULAR

Según Elrom (2016) “Originalmente llamado AngularJS, Angular es el marco de aplicaciones web front-end de open source JavaScript (basado en TypeScript) de Google. Está diseñado específicamente para crear aplicaciones web dinámicas. Con este marco, se puede desarrollar aplicaciones basadas en front-end sin tener que usar otros complementos o marcos. Además, se utiliza para desarrollar aplicaciones del lado del cliente de última generación, especialmente aplicaciones de una sola página.”

El mismo autor indica que es muy similar al marco de JavaScript, ya que es de código abierto. Además un grupo de desarrolladores de Google ha creado este marco de Angular, que es uno de los marcos más extendidos y modernos. También, cuenta con un gran apoyo del equipo de Angular CLI y Angular App e Google. Asimismo, este admite varias plataformas, como dispositivos nativos móviles, web y de escritorio. También se dispone de patrones de diseño para crear las tareas en un método más fácil para crear una aplicación. A continuación, según Elrom (2016) se presentan algunas de las funciones disponibles con Angular:

- **Menos código:** Se necesitan muchos scripts solo para diseñar una aplicación al realizar la manipulación DOM. Pero se requiere menos código para la manipulación de DOM si se usa Angular.
- **Controlador de vista de modelo:** El marco angular se construye sobre la base de Model-View-Controller (MVC).
- **Examen de la unidad:** Dispone de un excelente marco de prueba conocido como Karma diseñado por el equipo de desarrollo de Google. Esto es útil para realizar el diseño de pruebas unitarias para aplicaciones AngularJS.
- **Enlace del modelo de datos:** Al vincular datos a los controles HTML, no se requiere un código especial, solo agregando pocos fragmentos de código es posible vincular datos, lo que se hace a través de Angular.

- **Aplicaciones de escritorio:** Con Angular, se puede crear aplicaciones que se instalan directamente en el escritorio tales como Windows, Mac y Linux.
- **Desplazamiento virtual:** Para cargar y descargar elementos de DOM, se utiliza principalmente el desplazamiento virtual en Angular. Este proceso se basa completamente en las partes visibles de las listas.

2.4.6. POSTGRESQL

Es un sistema que permite una gestionar la diferente información que se almacena dentro de BD, una de mayores ventajas es que es de código abierto, es decir no es necesario de una licencia para poder hacer uso del mismo, de igual forma permite realizar sugerencias relacionales.

2.5. LECTORES Y HERRAMIENTAS DE EVALUACIÓN

2.5.1. Lectores de Pantalla

Es una tecnología que permite interactuar con la programación que esta implementada dentro del sistema una de ellas es siguiendo las normativas de WAI-ARIA.

2.5.1.1. NVDA(Non Visual Desktop Access)

NVDA (Non Visual Desktop Access) Es una tecnología que brinda la función como lector de pantalla gratuita diseñado por NV Access, tiene la posibilidad de leer toda la estructura de la tecnología de WAI-ARIA para los usuario con discapacidad llegando así a convertir en un herramienta indispensable para personas que tiene este tipo de problemas.

2.5.1.2. JAWS

Este sistema permite leer en voz alta el contenido de la pantalla del PC y tiene el beneficio que a los usuarios puede entender toda la información el mismo que podra realizar las acciones sin mayor dificultad.

2.5.2. Herramientas de Evaluación

Hoy en día existe un sin número de herramientas que me permiten realizar la evaluación a una página web permitiendo así llevar un control de monitoreo para cada una de las personas que tienen alguna discapacidad, a continuación presentamos algunos de ellos:

2.5.2.1. WAVE

Según WAVE (2022) “Son un grupo de funcionalidades que pueden emplear desarrolladores para evaluar si el contenido web es accesible para personas con discapacidad.” Como de igual forma, tiene un servicio WAVE Runner, API WAVE de suscripción y una API WAVE independiente para recopilar toda la información de forma rápida y eficiente dentro de muchas páginas, presentando incluso informes de nivel empresarial y seguimiento de la accesibilidad, WAVE impulsa la herramienta de accesibilidad Pope Tech.

2.5.2.2. IBM EQUAL ACCESS ACCESSIBILITY CHECKER

“Es una herramienta de código abierto para desarrolladores y auditores web que utiliza el motor de reglas de accesibilidad de IBM, que detecta los problemas de accesibilidad de las páginas y aplicaciones web. La extensión se integra en las herramientas de desarrollo del navegador, proporcionando una experiencia de comprobación integrada, que ayuda a los usuarios a identificar rápidamente el origen de los problemas de accesibilidad y a intentar solucionarlos.” (IBM, 2022)

Capítulo III

MARCO METODOLÓGICO

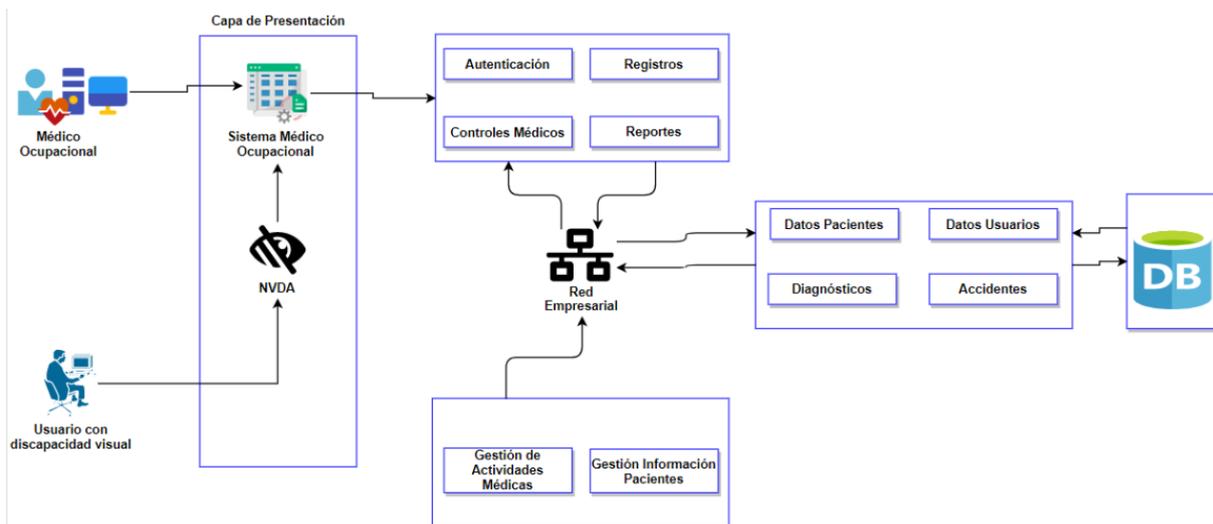


Figura 3.1: Arquitectura para Sistema Médico Ocupacional.

En la Figura se puede apreciar de una manera general las personas o usuarios presentes en el software, los distintos módulos que el sistema implementará y las relaciones que permitirán la comunicación entre ellos.

- **Capa de Presentación o Interfaz:** Esta capa permite la interacción con el usuario presentado en un formato legible toda la información que estará formado por dicho Sistema Médico.

- **Capa de Negocio:** En ella tenemos toda la parte lógica del programa o sistema permitiendo manejar toda la parte de la gestión de los pacientes como de igual forma de las distintas actividades y controles médicos a realizar.
- **Capa de Datos:** Encargada de persistir toda la información o datos del sistema en ella se almacenará todo lo relacionado a los datos de pacientes, datos de usuarios (Administrador-Médico, Enfermero) como los distintos Diagnósticos y Accidentes relacionados con los pacientes.

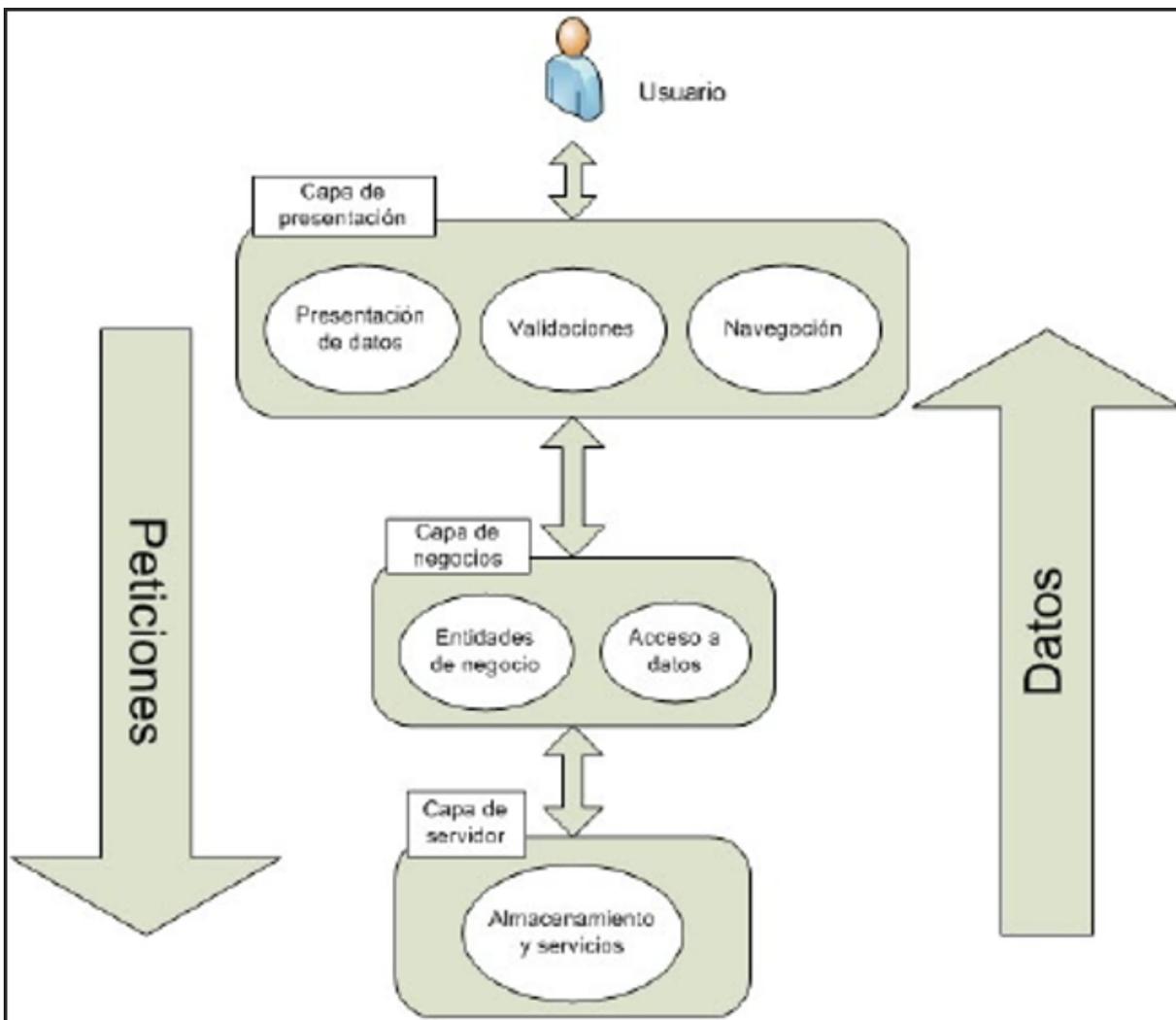


Figura 3.2: Esquema Lógico de las capas(Sacoto Tapia, 2020).

3.1. División de la capa de negocios

(Sacoto Tapia, 2020) “La capa de negocios representa el grueso de la lógica de funcionamiento de la aplicación distribuida.” Una de las características de esta capa como es de genocisos es aislar la capa de presentación de la capa de servidor, de forma que las estructuras de datos subyacentes y la lógica que las utilizan sean independientes de la capa de presentación. De esta forma, también, el mantenimiento de las normas de negocio será más sencillo y, sobre todo, será reutilizable desde cualquier capa de presentación, sea del tipo que sea.

3.1.1. Entidades de negocio

"Las entidades de negocio son estructuras de datos que la aplicación maneja y que representan a las entidades de datos definidas en los orígenes de datos. Una entidad de negocio tendrá elementos que se correspondan, en todo o en parte, con los elementos de la entidad de datos a la que representan. Por lo general las entidades de negocio no poseen métodos sino propiedades, ya que su finalidad es la de describir la entidad de negocio a la que representan."(Sacoto Tapia, 2020).

3.1.2. Objeto de acceso a datos

En esta parte lo que me permite es ser un intermediario lo que es entre la capa de aplicación con los orígenes de datos como puede ser por ejemplo de de una base de datos, el mismo que me delante ya sea api pueden llegara realizar dicha conexión, permitiendo realizar ya sea sentencias SQL para poder consular o guardar información.

Existen algunas características de métodos para lo que refiere al acceso a datos las cuales pueden ser:

- Métodos de procedimiento: Dentro de estos métodos podemos realizar lo que es conocido

por sus siglas CRUD, son un conjunto de instrucciones que mediante sentencias SQL se puede persistir datos, borrar, actualizar y listar toda la información que se desee es su momento.

- Método procedimiento: También conocidos como procedimientos almacenados, en donde se le puede realizar directamente en un gestor de BASE DE DATOS.

3.1.3. Objeto de negocio

"Los objetos de negocio son una abstracción de un conjunto de entidades de datos relacionadas entre sí. El objetivo de estos objetos es encapsular el acceso a varios objetos de acceso a datos en un único objeto, de modo que las capas superiores no hayan de realizar accesos a múltiples objetos para realizar una operación."(Sacoto Tapia, 2020).

3.1.4. Fachadas de negocio

"Las fachadas de negocio sólo aparecen en grandes aplicaciones con muchos objetos de acceso a datos y muchos objetos de negocio. En este tipo de aplicaciones será necesario encapsular algunos conjuntos de objetos de negocio en un objeto más complejo que encapsule ciertas operaciones a fin de que la programación de la capa de presentación sea más fácil y más coherente."(Sacoto Tapia, 2020).

3.2. Requerimientos

Los requerimientos para este desarrollo e implementación del presente proyecto son de tipo hardware y software los cuales permitan una correcta ejecución del sistema en el cual estos son importantes para el funcionamiento del mismo. Los cuales son detallados a continuación.

3.2.1. Hardware

- Servidor Web, Base de Datos

S.O. Ubuntu Server, servicios de apache2 y Postgresql.

3.2.2. Software

- JDK (Java Development Kit).

Un software que proporciona funciones y herramientas las mismas que son utilizadas para el desarrollo y creación de programas dentro de Java.

- JAVA.

Lenguaje de programación orientado a objetos el cual permite el desarrollo de aplicaciones de manera versátil y sencilla.

- Servidor de aplicaciones - WildFly

WildFly como tal es un servidor de aplicaciones compatible con JAVA JEE, también es Open Source es decir no tiene ningún costo alguno, teniendo así grandes ventajas como su flexibilidad, ligero y administrativo en la creación e implementación de aplicaciones y servicios JAVA.

- Angular

Librerías: Angular material, bootstrap.

- IDE desarrollo

Visual Studio Code: Extensiones - Tabnine, Terminal.

Eclipse JEE

- POSTGRESQL

pgAdmin III: Herramienta de gestión y administración para la base de datos relacionales.

3.3. Requerimientos del Aplicativo

Para el desarrollo del proyecto se dio inicio con el levantamiento de los requerimientos al formar parte de una organización como lo es Farmasol EP. Son indispensables para la correcta

realización del sistema.

Los requerimientos están realizados con énfasis en los requisitos del Médico Ocupacional para su correcta interacción con la plataforma, al estar plasmado los mismos según sus necesidades.

Finalmente, al haber culminado con el proceso de estudio y análisis de todos los requerimientos que se verán involucrados en el sistema, se pudo definir de manera más concisa y eficaz las funcionalidades que el sistema deberá realizar para su correcto funcionamiento.

3.3.1. Identificación de stakeholders

Dentro de un proyecto es imprescindible identificar los involucrados que interactuarán de manera directa o indirectamente en la realización del sistema, para lo cual se define a continuación:

- Dentro del grupo se encuentran los siguientes involucrados:
 - **Médico ocupacional:** Perteneciente a Farmasol EP el cual estará encargado de utilizar el sistema y corroborar el funcionamiento del mismo.
 - **Desarrolladores:** Encargados en el desarrollo de la plataforma, los cuales permitirán cumplir con los objetivos y requerimientos propuestos anteriormente.
 - **Gerencia de FARMASOL:** Los cuales nos brinda asistencia en la documentación para la realización de nuestro proyecto dentro de la Empresa Pública de Farmacias Municipales Solidarias FARMASOL EP.
 - **Departamento de Sistemas:** El cual nos ayuda con la gestión de hardware para el despliegue del software como tal.
 - **Pacientes o trabajadores:** Los cuales se les realizará los distintos tipos de consultas dependiendo su estado de salud.
 - Todas las pruebas con el grupo de involucrados, se las realizarán dentro de la Empresa Farmasol EP.

3.3.2. Requerimientos Funcionales

Estos requerimientos hacen una alusión a las acciones y servicios que debe proveer un sistema, habitualmente hacen referencia a las salidas y entradas de cada uno de los procesos y almacenamiento de la información dentro del sistema, los cuales se presenta a continuación:

Código	RF01
Nombre	Registro de Usuarios
Actor	Administrador
Descripción	- El sistema debera permitir registrar un usuario con un rol asignado.
Proceso	Registrar Usuario y contraseña. - Asignar un ROI. - Sistema Verifica los campos ingresados correctamente. -Usuario registrado.
Post condiciones	Validar la información ingresada. Registro guardado correctamente. Si el registro no se guarda se emite un mensaje de error
Prioridad	ALTA

Tabla 3.1: Requisito Funcional 1

Código	RF01
Nombre	Lista de Usuario
Actor	Administrador
Descripción	El sistema permite listar usuarios con su respectivo rol.
Proceso	- El Administrador ingresa al módulo Usuarios. - El Administrador visualiza la información de los usuarios.
Post condiciones	El sistema emitirá una lista que incluya toda la información del usuario con su respectivo rol, permitiendo identificar el mismo.
Prioridad	MEDIA

Tabla 3.2: Requisito Funcional 1

Código	RF02
Nombre	Registrar Paciente
Actor	Doctor
Descripción	- El sistema deberá permitir registrar un paciente con un departamento asignado.
Proceso	Registrar Paciente. - Asignar un Departamento. - Sistema Verifica los campos ingresados correctamente. -Paciente registrado.
Post condiciones	Validar la información ingresada. Si el registro no se guarda se emite un mensaje de error
Prioridad	ALTA

Tabla 3.3: Requisito Funcional 2

Código	RF03
Nombre	Registrar Examen Inicial
Actor	Doctor
Descripción	- El sistema debera permitir registrar un Examen Inicial asignado a un paciente.
Proceso	Registrar Examen Inicial. - Seleccionar paciente para Examen Inicial. - Sistema Verifica los campos ingresados correctamente. - Consulta Examen Inicial.
Post condiciones	Validar la información ingresada. Si el registro no se guarda se emite un mensaje de error
Prioridad	ALTA

Tabla 3.4: Requisito Funcional 3

Código	RF04
Nombre	Registrar Consulta Medica
Actor	Doctor
Descripción	- El sistema debera permitir registrar una consulta medica asignado a un paciente.
Proceso	Registrar Consulta medica. - Seleccionar paciente para ingresar una consulta medica. - Sistema Verifica los campos ingresados correctamente. - Consulta Medica registrada.
Prioridad	ALTA
Post condiciones	Validar la información ingresada. Si el registro no se guarda se emite un mensaje de error
Prioridad	ALTA

Tabla 3.5: Requisito Funcional 4

Código	RF05
Nombre	Listar Consultas
Actor	Doctor
Descripción	- El sistema permitira listar todas las consultas de cualquier tipo ya sea medica, preocupacional, entre otros, pertenecientes a un paciente.
Proceso	- Ingreso al modulo de consulta ya sea media, preocupacional, etc. - Seleccionar paciente para listado de consultas. - Listar todas las consultas que un paciente se a realizado.
Post condiciones	Visualizar la información de las consultas.
Prioridad	ALTA

Tabla 3.6: Requisito Funcional 5

Código	RF05
Nombre	Registrar Consulta Preventiva
Actor	Doctor
Descripción	- El sistema debera permitir registrar una Consulta Preventiva asignado a un paciente.
Proceso	Registrar Consulta Preventiva. - Seleccionar paciente para ingresar una consulta preventiva. - Sistema Verifica los campos ingresados correctamente. - Consulta preventiva registrada.
Post condiciones	Validar la información ingresada. Si el registro no se guarda se emite un mensaje de error
Prioridad	ALTA

Tabla 3.7: Requisito Funcional 6

Código	RF06
Nombre	Registrar Consulta Preocupacional
Actor	Doctor
Descripción	- El sistema debera permitir registrar una Consulta Preocupacional asignado a un paciente.
Proceso	Registrar Consulta Preocupacional. - Seleccionar paciente para ingresar una consulta preventiva. - Sistema verifica los campos ingresados correctamente. - Consulta Preocupacional registrada.
Post condiciones	Validar la información ingresada. Si el registro no se guarda se emite un mensaje de error
Prioridad	ALTA

Tabla 3.8: Requisito Funcional 7

Código	RF07
Nombre	Registrar Examen Medico Inicial
Actor	Doctor
Descripción	- El sistema debera permitir registrar un Examen Medico Inicial asignado a un paciente.
Proceso	Registrar Examen Medico Inicial. - Seleccionar paciente para ingresar una Examen Medico Inicial. - Sistema verifica los campos ingresados correctamente. - Examen Medico Inicial registrada.
Post condiciones	Validar la información ingresada. Si el registro no se guarda se emite un mensaje de error
Prioridad	ALTA

Tabla 3.9: Requisito Funcional 8

Código	RF08
Nombre	Registrar Examen Medico Retiro
Actor	Doctor
Descripción	- El sistema debera permitir registrar un Examen Medico Retiro asignado a un paciente.
Proceso	Registrar Examen Medico Retiro. - Seleccionar paciente para ingresar una Examen Medico Retiro. - Sistema verifica los campos ingresados correctamente. - Examen Medico Retiro registrada.
Post condiciones	Validar la información ingresada. Si el registro no se guarda se emite un mensaje de error
Prioridad	ALTA

Tabla 3.10: Requisito Funcional 9

Código	RF09
Nombre	Ficha Reintegro
Actor	Doctor
Descripción	- El sistema debera permitir registrar una Ficha Reintegro asignado a un paciente.
Proceso	Registrar Ficha Reintegro. - Seleccionar paciente para ingresar una Ficha Reintegro. - Sistema verifica los campos ingresados correctamente. - Ficha Reintegro registrada.
Post condiciones	Validar la información ingresada. Si el registro no se guarda se emite un mensaje de error
Prioridad	ALTA

Tabla 3.11: Requisito Funcional 10

Código	RF10
Nombre	Registrar Accidente
Actor	Doctor
Descripción	- El sistema debera permitir registrar una Accidente asignado a un paciente.
Proceso	Registrar Accidente. - Seleccionar paciente para ingresar una Accidente. - Sistema verifica los campos ingresados correctamente. - Accidente registrada.
Post condiciones	Validar la información ingresada. Si el registro no se guarda se emite un mensaje de error
Prioridad	ALTA

Tabla 3.12: Requisito Funcional 11

Código	RF11
Nombre	Registrar Historial
Actor	Doctor
Descripción	- El sistema debera permitir registrar una Historial asignado a un paciente.
Proceso	Registrar Historial. - Seleccionar paciente para ingresar una Historial. - Sistema verifica los campos ingresados correctamente. - Historial registrada.
Post condiciones	Validar la información ingresada. Si el registro no se guarda se emite un mensaje de error
Prioridad	ALTA

Tabla 3.13: Requisito Funcional 12

Código	RF12
Nombre	Reportes
Actor	Doctor
Descripción	- El sistema permitira imprimir reporte en pdf las consultas registradas por fecha.
Proceso	- Ingreso de fecha desde hasta para visualizar e exportar en PDF. - Visulaizar Reportes.
Post condiciones	Validar fechas ingresadas. presentación de los datos obtenido en un PDF
Prioridad	ALTA

Tabla 3.14: Requisito Funcional 13

3.3.3. Requerimientos No Funcionales

Código	RNF01
Nombre	Interrelación el usuario con Software
Descripción	El usuario podra utilizar de la manera mas sencilla el software.
Características	Interfaz sencilla y accesible para el usuario
Prioridad	MEDIA

Tabla 3.15: Requisito No Funcional 1

Código	RNF02
Nombre	Integridad de datos
Descripción	Cifrar la información.
Características	Correctitud y completitud de la información almacenada en BD.
Prioridad	ALTA

Tabla 3.16: Requisito No Funcional 2

Código	RNF03
Nombre	Elección de un Sistema Operativo
Descripción	Analizar a detalle de todas las especificaciones y requisitos del software para tener un buen rendimiento en la ejecución del sistema.
Características	Obtener todos los detalles del software para del despliegue del mismo.
Prioridad	ALTA

Tabla 3.17: Requisito No Funcional 3

Código	RNF04
Nombre	Recursos para despliegue del software
Descripción	Investigar con detalle las especificaciones de la plataforma para el desarrollo.
Características	Almacenamiento, rendimiento
Prioridad	ALTA

Tabla 3.18: Requisito No Funcional 4

Código	RNF05
Nombre	Accesibilidad
Descripción	Tamaño de letra y colores.
Características	- El tamaño de la letra debe ser mínimo 14. - El color deberá mantener un contraste entre la letra y el fondo.
Prioridad	ALTA

Tabla 3.19: Requisito No Funcional 5

3.4. Diagrama para la aplicación Web

3.4.1. Diagrama Entidad-Relación

Es sumamente importante para llegar a identificar el proceder de cada una de las clases que están asociadas que participan de una u otra forma en el sistema.

Como se puede apreciar en la Figura 3.3, el sistema contiene un sin número de tablas pero las que más se recalcan a nivel de de interfaz gráfica son las siguientes:

Tabla de Usuarios: La cual se encargara de guardar la información de un usuario con su rol respectivo.

La tabla consulta : Ingresando la información de la consulta de un paciente.

La tabla Accidentes : Ingresando la información de un accidente a un paciente.

3.5. Desarrollo

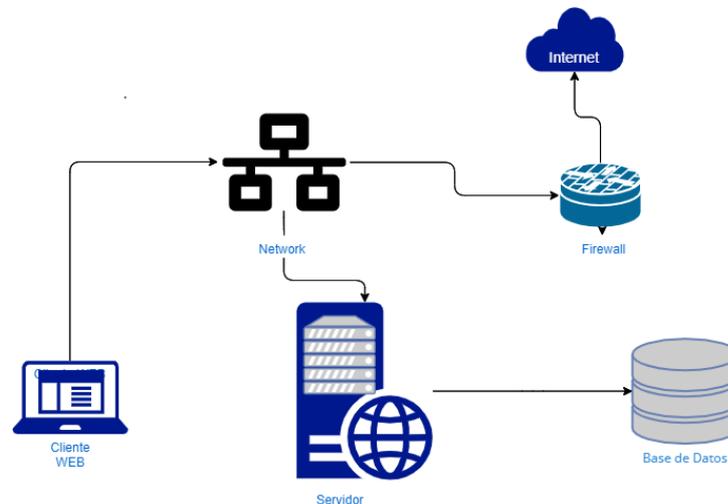


Figura 3.5: Diagrama físico.

Para el desarrollo del proyecto, se ha decidido implementar una arquitectura N-capas debido a la robustez, seguridad, confiabilidad y escalabilidad que ofrece este tipo de arquitecturas debido a que cada una de las capas se encuentran separadas permitiendo una mejor administración en cada una de ellas, utilizando tecnología JEE para la implementación del backend.

Para el almacenamiento de la información se optó por implementar un gestor de BD objeto-relacional como lo es PostgreSQL, debido a una de sus grandes ventajas que es de código abierto, de igual forma siendo la más adecuada para proyectos que manejen un gran volumen de datos.

3.5.1. SERVIDOR JAVA

El modelo clásico de capas en la arquitectura de las aplicaciones Java EE se divide en las siguientes:

- **Cliente:** Normalmente se trata de un navegador, pero puede ser un teléfono inteligente.

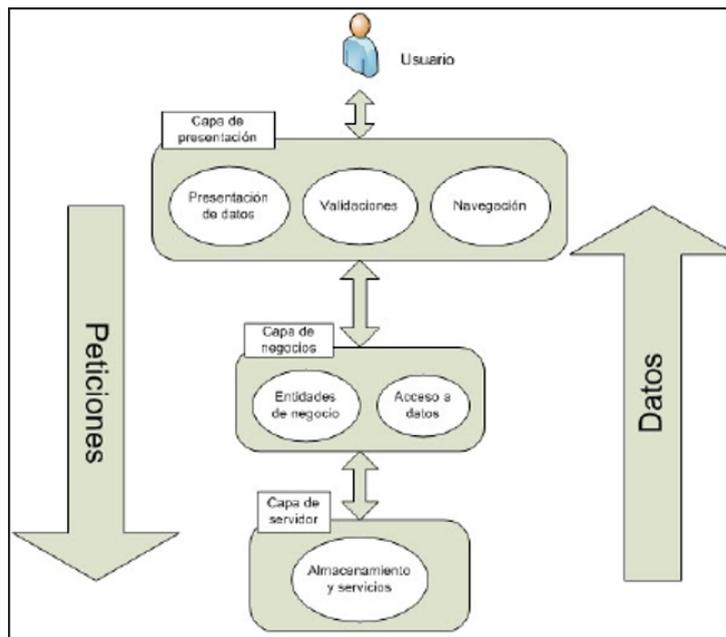


Figura 3.6: Esquema Lógico de las capas(Sacoto Tapia, 2020).

- **Capa web:** Permite la intercomunicación lo que es entre cliente con la capa de negocios permitiendo abstraer toda la información necesaria que el cliente lo este solicitando.
- **Capa de negocio:** En esta parte permite realizar todo el procedimiento para realizar lo que es la persistencia de toda la información.
- **Capa del servidor:** Permite persistir los datos que un usuario pretende guardar para luego ser recuperada en cualquier momento, BD como PostgreSQL, oracle, etc.

Para la implementación del servidor se usó **JAVA** utilizándolo en el Entorno de Desarrollo Eclipse JEE en su versión 2021 para lo cual se procedió a instalar un servidor de aplicaciones como lo es JBoss específicamente en la versión 4.21.3 Final siendo la versión mas estable existente hasta ahora, necesario ya que brinda soporte JPA y muchas otras herramientas.

3.5.1.1. Wildfly

Según WildFly (2022) “Es un servidor de aplicaciones con certificación Java EE 8.”

Entre los servicios que ofrece se pueden describir los siguientes:

- Un grupo de conexiones JDBC para una administración a una BD relacional, el grupo de conexiones se puede usar directamente al inyectar un objeto Datasource en su aplicación o indirectamente usando, por ejemplo, una aplicación JPA que se basa en un Datasource disponible.
- Un contenedor EJB donde puede ser acabo implementaciones de servicios remotos como por ejemplo los EJB.
- Un servidor web ligero y rápido llamado "Undertow". Este servidor web brinda soporte para aplicaciones web heredadas (JSP / Servlets), así como para interfaces JSF y RESTful.
- "Wildfly es ahorra tiempo: Su proceso de arranque está altamente optimizado, los servicios se inician simultáneamente eliminando las esperas."(genos, 2021).
- "Ahorro de memoria. Su gestión de la memoria es capaz de minimizar la asignación de posiciones, evitando la carga de clases duplicadas y objetos, aligerando también los procesos del colector de basura de Java."(Arsys, 2017).
- "Basado en estándares. Aprovecha las últimas mejoras de Java EE7 para maximizar las capacidades de las aplicaciones y la productividad de los desarrolladores. Además, dispone de una plataforma adaptada a todos los últimos estándares de la web moderna y el acceso desde dispositivos. Soporta REST, JAX-RS 2 y JSON-P, entre otros."(Arsys, 2017)

El servidor de aplicaciones lo podemos instalar directamente de la tienda haciendo uso del IDE de Desarrollo.

Según (genos, 2021) "Indica que, con la intención de satisfacer la demanda de conectividad, capacidad de respuesta y capacidad de escalar, que surgen en el desarrollo de aplicaciones webs."

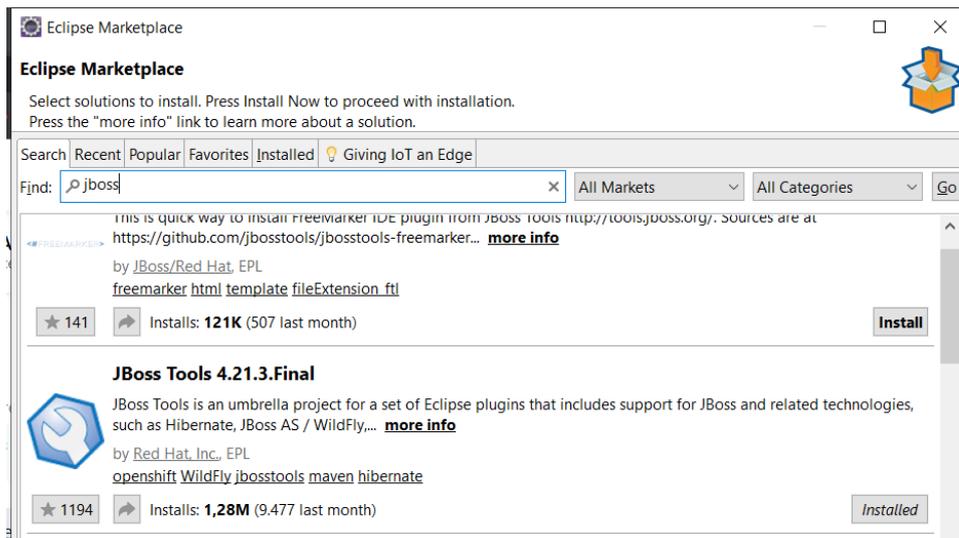


Figura 3.7: Servidor de Aplicaciones

Así el servidor permitirá alojar nuestra aplicación y proporcionar el entorno para la ejecución de la misma.

Es importante para empezar con el desarrollo configurar el archivo standalone.xml en nuestro servidor de aplicaciones para permitir una comunicación con nuestra base de datos POSTGRESQL y así poder persistir toda nuestra información. En la siguiente imagen se muestra el DataSource asignado para la respectiva conexión desde el servidor hacia la Base de Datos especificada.

```

<datasources>
  <datasource jndi-name="java:jboss/datasources/ExampleDS" pool-name="ExampleDS" enabled="true" use-java-context="true" statistics-enabled="${wildfly.d
  <connection-uri>|jdbc:h2:mem:test;DB_CLOSE_DELAY=1;DB_CLOSE_ON_EXIT=FALSE</connection-uri>
  <driver>h2</driver>
  <security>
    <user-name>sa</user-name>
    <password>sa</password>
  </security>
</datasource>
  <datasource jta="true" jndi-name="java:jboss/datasources/ejPostgresDS" pool-name="ejPostgresDS" enabled="true" use-java-context="true" use-ccm="true"
  <connection-uri>|jdbc:postgresql://localhost:5432/postgres</connection-uri>
  <driver>postgresql</driver>
  <security>
    <user-name>postgres</user-name>
    <password>postgres</password>
  </security>
</datasource>
  <driver name="h2" module="com.h2database.h2">
    <xa-datasource-class>org.h2.jdbc.JdbcDataSource</xa-datasource-class>
  </driver>
  <driver name="postgresql-driver" module="org.postgresql">
    <driver-class>org.postgresql.Driver</driver-class>
  </driver>
</drivers>
</datasources>

```

Figura 3.8: Configuración DataSource

El DataSource contiene los siguientes parámetros que deben modificarse dependiendo de la base de datos a emplearse los demás los podremos dejar por defecto:

- **Connection-url:** Para la conectividad de la base de datos java agregando el puerto y nombre de la BD a conectarse.
- **User-name:** Nombre del usuario para BD.
- **Password:** Credenciales para la conexión a una BD.

Para poder utilizar todas las librerías necesarias en un proyecto **JAVA EE** se las puede importar directamente desde el archivo llamado "pom.xml" tanto librerías para la persistencia de datos como lo es JPA como el Driver JDBC de postgres necesarios para este proyecto.

Se puede observar que el servidor de aplicaciones en si viene incluido con las librerías necesarias para hacer uso de la API de persistencia de Java.

```

<!-- Import the CDI API, we use provided scope as the API is included in
JBoss WildFly -->
<dependency>
<groupId>javax.enterprise</groupId>
<artifactId>cdi-api</artifactId>
<scope>provided</scope>
</dependency>

<!-- Import the Common Annotations API (JSR-250), we use provided scope
as the API is included in JBoss WildFly -->
<dependency>
<groupId>org.jboss.spec.javax.annotation</groupId>
<artifactId>jboss-annotations-api-1.2_spec</artifactId>
<scope>provided</scope>
</dependency>

<!-- Import the JAX-RS API, we use provided scope as the API is included
in JBoss WildFly -->
<dependency>
<groupId>org.jboss.resteasy</groupId>
<artifactId>jaxrs-api</artifactId>
<scope>provided</scope>
</dependency>

<!-- Import the JPA API, we use provided scope as the API is included in
JBoss WildFly -->
<dependency>
<groupId>org.hibernate.javax.persistence</groupId>
<artifactId>hibernate-jpa-2.1-api</artifactId>
<scope>provided</scope>
</dependency>

```

Figura 3.9: Archivo pom.xml

Dentro del servidor de aplicaciones tenemos la opción de vincular una dirección en particular utilizada para que cualquier equipo en la red pueda acceder al servidor y obtener comunicación hacia los distintos servicios creados para la aplicación siendo muy importante en el momento que procedamos a desplegar nuestro sistema.

El parámetro importante a configurar es el parámetro `-bind` en el cual especificaremos la dirección IP a vincular al servidor.

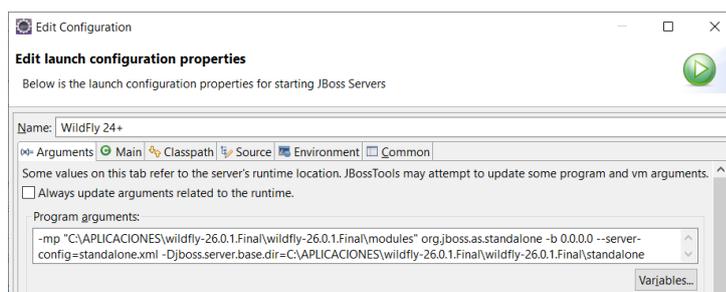


Figura 3.10: Configuración de las propiedades del servidor

3.5.1.2. CREACIÓN DE SERVICIOS REST

“REST no es un protocolo ni un estándar, sino más bien un conjunto de límites de arquitectura.” (Hat, 2020) El servicio REST es una de las api mas utilizada por los desarrolladores ya que permite la intercomunicación con la capa del servidor, presentando los datos en un formatos

ya sea JSON el mismo que mediante plataformas de desarrollo podemos implementar métodos que nos permitan ya sea mostrar los datos o enviar los datos desde el usuario final.

MÉTODOS API REST

- **POST:** Mediante este método nos permite almacenar la información dentro de una Base de Datos enviados desde nuestra interfaz de usuario.
- **PUT:** Mediante este método nos permite realizar una modificación ya sea a todo un objeto o partes de las mismas teniendo en cuenta que ya estaba previamente creado.
- **GET:** Con este método lo que aremos es, realizar una petición a nuestro servidor de una cierta información que este optando el usuario.
- **DELETE:** Método el cual tiene como funcionalidad eliminar un objeto oh cierta información.

Para que nuestro Front-end pueda acceder a los datos almacenados y pueda realizar todas las funcionalidades asignadas al sistema como Crear, Leer, Actualizar y Eliminar información como al acceso de lo que respecta a la lógica de negocio se desarrollaron varios servicios rest con sus respectivos métodos y parámetros de entrada visualizando los principales servicios a continuación:

Registrar Paciente

Parámetros de entrada

Paciente: Objeto de tipo paciente para el registro.

```

@POST
@Path("/registrarPaciente")
@Consumes("application/json")
@Produces("application/json")
public Response insertPaciente(Paciente paciente) {
    Response.ResponseBuilder builder=null;
    Map<String, String> data=new HashMap<>();

    try {
        pon.save(paciente);
        data.put("code", "1");
        data.put("message", "OK guardado");
        builder=Response.status(Response.Status.OK).entity(data);
    }
    catch(Exception e) {
        data.put("code", "99");
        data.put("message", e.getMessage());
        builder=Response.status(Response.Status.OK).entity(data);

        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    return builder.build();
}

```

Figura 3.11: Método registrar Paciente

Lista Pacientes

Parámetros de entrada

Paciente: Objeto paciente.

```

@GET
@Path("/listarPacientes")
@Produces("application/json;charset=utf-8")
public List<Paciente> getPacientes() {
    return pon.getPacientes();
}

```

Figura 3.12: Método listar Pacientes

Registrar Consulta Médica

Parámetros de entrada

Consulta Medica: Objeto de tipo consulta médica para el registro.

```
@POST
@Path("/registrarConsultaMedica")
@Consumes("application/json")
@Produces("application/json")
public Response registrarConsultaMedica(ConsultaMedica consultaMedica) {
    Response.ResponseBuilder builder=null;
    Map<String, String> data=new HashMap<>();

    try {
        con.saveConsultaMedica(consultaMedica);
        data.put("code", "1");
        data.put("message", "OK guardado consulta medica");
        builder=Response.status(Response.Status.OK).entity(data);
    } catch (Exception e) {
        // TODO: handle exception
        data.put("code", "99");
        data.put("message", e.getMessage());
        builder=Response.status(Response.Status.OK).entity(data);

        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return builder.build();
}
```

Figura 3.13: Método registrar Consulta Médica

Lista Consultas Médicas Por Paciente

Parámetros de entrada

Codigo: Codigo del paciente para visualizar sus consultas.

```
@GET
@Path("/listaConsultasMedicasPorPaciente")
@Produces("application/json;charset=utf-8")
public List<ConsultaMedica>getConsultaMedicaPorPaciente(@QueryParam("codigo")Integer codigo) {

    return con.getConsultasMedicasPorPaciente(codigo);
}
```

Figura 3.14: Método Listar Consultas Médica

Obtener Consulta Médica por Código

Parámetros de entrada

Código: Código de la consulta a actualizar datos.

```
@GET
@Path("/consultaMedicaPorCodigo")
@Produces("application/json;charset=utf-8")
public ConsultaMedica getConsultaPorCodigo(@QueryParam("codigo")Integer id) {
    return con.getConsultaMedicaPorCodigo(id);
}
```

Figura 3.15: Método obtener Consulta Médica

Actualizar Consulta Médica

Parámetros de entrada

Consulta Medica: Objeto de tipo consulta médica a actualizar datos.

```
@PUT
@Path("/actualizarConsultaMedica")
@Consumes("application/json")
@Produces("application/json")
public Response mergeConsultaMedica(ConsultaMedica consultaMedica) {
    Response.ResponseBuilder builder=null;
    Map<String, String> data=new HashMap<>();

    try {
        con.updateConsultaMedica(consultaMedica);
        data.put("code", "1");
        data.put("message", "OK ACTUALIZADO CONSULTA MEDICA");
        builder=Response.status(Response.Status.OK).entity(data);
    }
    catch(Exception e) {
        data.put("code", "99");
        data.put("message", e.getMessage());
        builder=Response.status(Response.Status.OK).entity(data);

        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    return builder.build();
}
```

Figura 3.16: Método Actualizar Consulta Médica

Eliminar Consulta Médica

Parámetros de entrada

Código: Código de la consulta para la eliminarla del registro.

```
@DELETE
@Path("/eliminarConsultaMedica")
@Produces("application/json")
@Consumes("application/json")
public Response eliminarConsultaMedica(@QueryParam("codigo")Integer id) {
    Response.ResponseBuilder builder= null;
    Map<String, String> data=new HashMap<>();

    try {
        con.deleteConsultaMedicaPorId(id);
        data.put("code", "1");
        data.put("message", "OK ELIMINADA CONSULTA MEDICA");
        builder=Response.status(Response.Status.OK).entity(data);
    } catch (Exception e) {
        data.put("code", "99");
        data.put("message", e.getMessage());
        builder=Response.status(Response.Status.OK).entity(data);

        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return builder.build();
}
```

Figura 3.17: Método Eliminar Consulta Médica

Los servicios para los módulos referentes a Consultas y Accidente se manejan con la misma lógica como esta implementada la Consulta Médica recibiendo como parámetros según el tipo de Objeto.

Registrar Usuario

Parámetros de entrada

Usuario: Objeto de tipo usuario para el registro.

```
@POST
@Path("/registrarUsuario")
@Consumes("application/json")
@Produces("application/json")
public Response registrarUsuario(Usuario usuario) {
    Response.ResponseBuilder builder=null;
    Map<String, String> data=new HashMap<>();

    try {
        uon.save(usuario);
        data.put("code", "1");
        data.put("message", "OK guardado Uusuario");
        builder=Response.status(Response.Status.OK).entity(data);
    }
    catch(Exception e) {
        data.put("code", "99");
        data.put("message", e.getMessage());
        builder=Response.status(Response.Status.OK).entity(data);

        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    return builder.build();
}
```

Figura 3.18: Método Registrar Usuario

Get Usuarios

```
@GET
@Path("/listaUsuarios")
@Produces("application/json; charset=utf-8")
public List<Usuario> getUsuarios(){
    return uon.getUsuarios();
}
```

Figura 3.19: Método Listar Usuarios

Get Usuario PorCodigo

Parámetros de entrada

Código: Código de Usuario para actualizar datos.

```
@GET
@Path("/usuarioPorCodigo")
@Produces("application/json;charset=utf-8")
public Usuario getUsuarioPorCodigo(@QueryParam("codigo")Integer codigo) {
    return uon.getUsuarioPorCodigo(codigo);
}
```

Figura 3.20: Método Obtener Usuario

Borrar Usuario

Parámetros de entrada

Código: Código de Usuario para la eliminarlo del registro.

```
@DELETE
@Path("/eliminarUsuario")
@Produces("application/json")
@Consumes("application/json")
public Response borrarUsuario(@QueryParam("codigo")Integer id) {
    Response.ResponseBuilder builder= null;
    Map<String, String> data=new HashMap<>();

    try {
        uon.deleteid(id);
        data.put("code", "1");
        data.put("message", "OK ELIMINADO USUARIO");
        builder=Response.status(Response.Status.OK).entity(data);
    } catch (Exception e) {
        data.put("code", "99");
        data.put("message", e.getMessage());
        builder=Response.status(Response.Status.OK).entity(data);

        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return builder.build();
}
```

Figura 3.21: Método Eliminar Usuario

Todas las pruebas con este servidor fueron realizadas de manera local permitiendo así verificar su correcto funcionamiento y corregir errores que vayan surgiendo para posteriormente concluido todo el desarrollo del sistema poder desplegarlo y ponerlo en producción permitiendo así cumplir con las funcionalidades propuestas.

3.5.2. Creación de la Base de Datos

Para almacenar los datos necesitamos de un Sistema de que gestione toda nuestra información es por ello que utilizamos un BD en el cual pueda residir toda la información y así atender a las peticiones que pueda realizar los usuarios desde un navegador.

Para ello se empleó la base de datos POSTGRESQL por su gran capacidad para trabajar con un gran volumen de datos y otras ventajas más explicadas anteriormente además de acoplarse muy bien con el servidor de aplicaciones.

A continuación, según PostgreSQL (2022) se presentan las principales ventajas o beneficios de PostgreSQL:

- Puede ejecutar sitios web dinámicos y aplicaciones web como una opción de pila LAMP.
- El registro de escritura anticipada de PostgreSQL lo convierte en una base de datos altamente tolerante a fallas.
- El código fuente de PostgreSQL está disponible gratuitamente bajo una licencia de código abierto. Esto le permite la libertad de usarlo, modificarlo e implementarlo según las necesidades de su negocio.
- Admite objetos geográficos, por lo que puede usarlo para servicios basados en la ubicación y sistemas de información geográfica.
- Fácil de usar, bajo mantenimiento y administración para el uso integrado y empresarial de PostgreSQL.

Para ello haremos uso de la herramienta PGADMIN creación de una BD a usar en el desarrollo del proyecto y así gracias a JPA (JAVA PERSISTENCE API) a través de su objeto EntityManager poder administrar todas las entidades.

Al momento de ingresar a la plataforma de administración PGADMIN se nos muestra una pantalla..

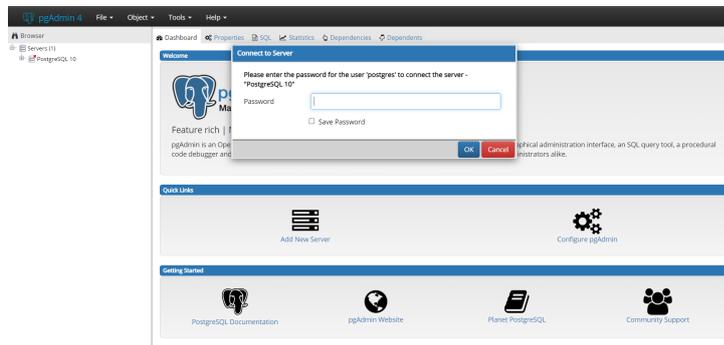


Figura 3.22: Pantalla inicio PGAdmin

Como se aprecia en la imagen debemos ingresar nuestra contraseña para nuestro usuario "**postgres**" y así poder acceder directamente a la consola de administración, allí encontramos el apartado "**Database**" donde podremos crear nuestra BD de igual forma el usuario destinado a administrar la misma.

Luego de haber creado nuestra base ya podremos hacer uso de ella a través de las configuraciones anteriormente mencionadas en el servidor de aplicaciones y contar así con un sistema para almacenar todos nuestros datos.

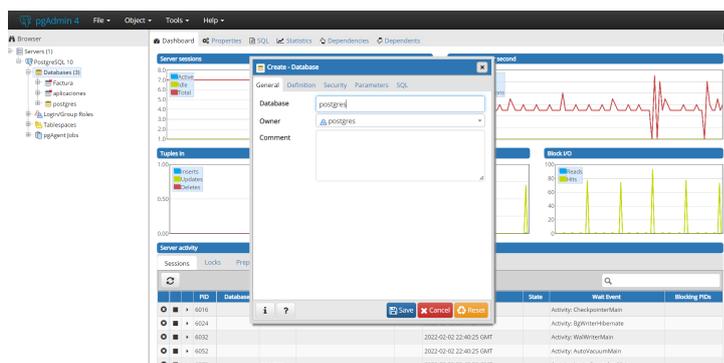


Figura 3.23: Consola de Administración PGAdmin

3.5.3. Etapa de Desarrollo Web

3.5.3.0.1 Angular 12

Es un Framework desarrollado por la empresa de Google mediante el lenguaje de TypeScript el mismo que me permite construir aplicaciones en HTML y JavaScript de una forma rápida y sencilla; es una plataforma dinámica y asíncrona con el servidor el mismo que me permite separar nuestro proyecto en apartado de Frontend y Backend de una forma totalmente separada.

3.5.3.0.2 TypeScript

"TypeScript define un conjunto de tipos de JavaScript, lo que ayuda a los usuarios a escribir código JavaScript que es más fácil de entender. Todo el código TypeScript se compila con JavaScript y se puede ejecutar sin problemas en cualquier plataforma. TypeScript no es obligatorio para desarrollar una aplicación Angular. Sin embargo, es muy recomendable ya que ofrece una mejor estructura sintáctica, al tiempo que hace que la base de código sea más fácil de entender y mantener"(Gonçalves, 2021).

3.5.3.0.3 Testing (Pruebas)

“Angular usa el Framework de prueba Jasmine. Jasmine proporciona múltiples funcionalidades para escribir diferentes tipos de casos de prueba.” (Gonçalves, 2021)

3.5.3.0.4 Módulos

En una app de Angular tiene un módulo raíz, el mismo que tiene como nombre AppModule en el cual suministra un herramienta para dar inicio a la aplicación.

3.5.3.0.5 Componentes

En la cual son los fragmentos de código en la cual define una clase dentro de Angular en la cual determina una lógica y los datos de un aplicativo.

3.5.3.0.6 Arquitectura de Angular

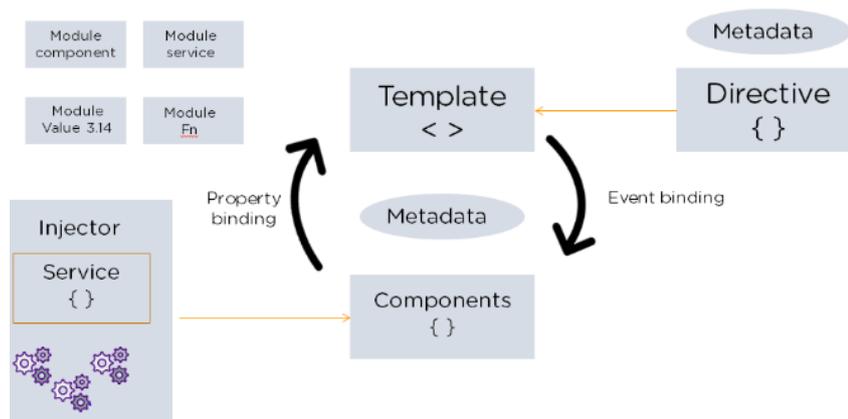


Figura 3.24: Arquitectura en Angula (Gonçalves, 2021).

El Framework como es Angular trabaja como Backend y de forma independiente el mismo que se encarga de realizar un HTTP Request hacia el Frontend y devuelve un HTTP Response que será percibida por el usuario final.

■ Velocidad y Rendimiento

- - Multiplataforma o aprendizaje.
- - Optimización de código.

• Productividad

- - Creación de componentes, módulos, etc.
- - Angular CLI.
- - Incorporación con la gran mayoría de editores de texto.

3.5.4. Descargar de Node.js y NPM

3.5.4.0.1 Prerequisitos

Para nuestro proyecto vamos a descargar los files de instalación de 64 bits para Node.js.

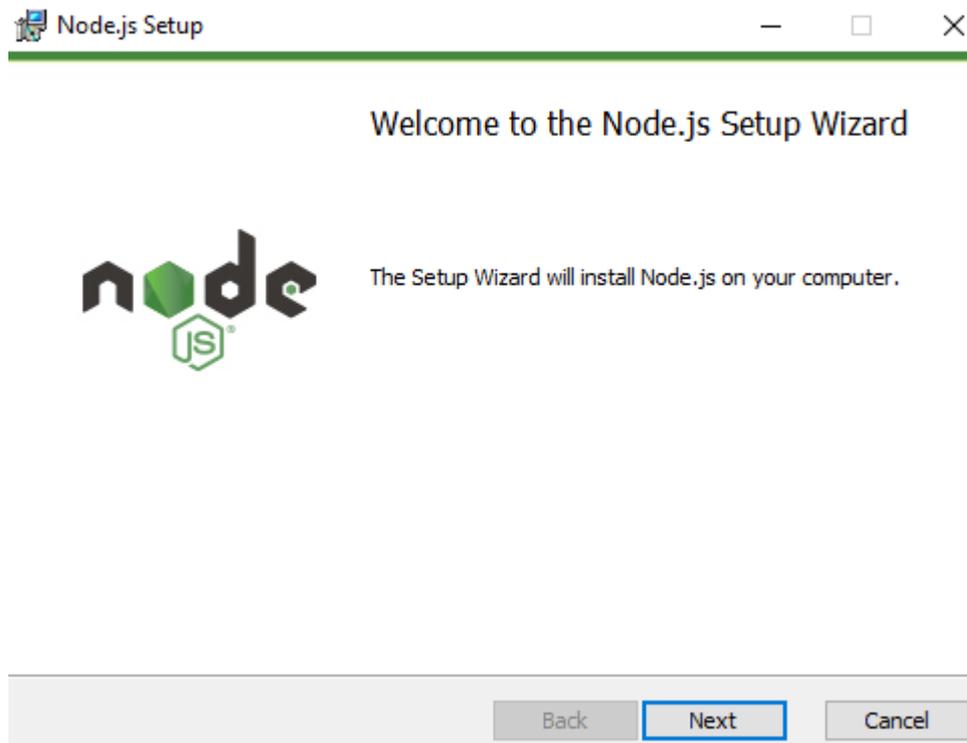


Figura 3.25: Instalación de Node js.

Para verificar que todo este correctamente instalado ingresmos los siguientes comandos dentro de nuestra terminal.

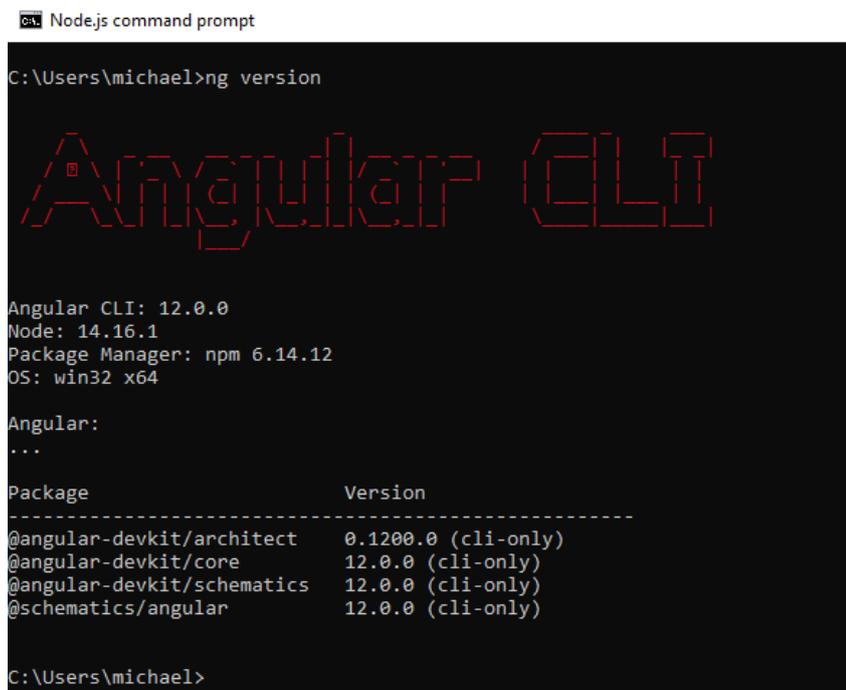
```
ca. Select Node.js command prompt
C:\Users\michael>node -v
v14.16.1
C:\Users\michael>npm -v
6.14.12
C:\Users\michael>
```

Figura 3.26: Instalación de Node js. cmd

3.5.4.0.2 Instalación de Angular CLI

Angular CLI, herramienta que permite crear proyectos, originar componentes, depurar y publicar, además de ello es potente, versátil y muy sencillo de interactuar con aspectos básico que tiene el mismo.

- *npm install -g @angular/cli.*



```
Node.js command prompt
C:\Users\michael>ng version

Angular CLI
Angular CLI: 12.0.0
Node: 14.16.1
Package Manager: npm 6.14.12
OS: win32 x64

Angular:
...

Package      Version
-----
@angular-devkit/architect    0.1200.0 (cli-only)
@angular-devkit/core        12.0.0 (cli-only)
@angular-devkit/schematics   12.0.0 (cli-only)
@schematics/angular         12.0.0 (cli-only)

C:\Users\michael>
```

Figura 3.27: Intalación de Angular Cli

3.5.4.0.3 Interfaz de desarrollo o IDE

Como ID de desarrollo se opto por Visual Studio Code, ya que una de sus ventajas es que permite una ejecución en multiplataforma; es un editor de código, también consiente en poder desarrollar software con diferentes lenguajes de programación, también permite gestionar

atajos personalizados y factorizar el código, VSC es gratuito y de código abierto, este IDE proporciona una gran variedad de extensiones en el cual es de gran ayuda, permitiendo optimizar un tiempo considerable al desarrollador, entonces podemos decir que este software es uno de los más usados por su flexibilidad, versatilidad y usabilidad que tiene el mismo.

3.5.5. Creación de Proyecto en Angular

Una vez descargado todas las herramientas necesarias, le damos continuidad a nuestro proyecto procediendo a la creación de un proyecto dentro de Angular, para ello:

- - Creamos una carpeta donde se guardará nuestro proyecto.
- - Abrimos la terminal de Node.js.
- - Una vez dentro del terminal nos situamos dentro de la carpeta creada anteriormente, para ello podemos utilizar el comando: `cd (Nombre de la carpeta)`.
- - La creación del proyecto como tal se realiza con el siguiente comando:

ng new SistemaMedicoOcupacionaApp

Este comando nos ayuda con la creación de la carpeta para nuestra aplicación en mismo que guardara cada uno de los archivos correspondientes para la correcta ejecución del mismo.

3.5.5.0.1 Instalación de librerías Angular

- ***BOOTSTRAP***

Es un framework front-end con una característica como es de código abierto, disponible para CSS y HTML el mismo que también brinda una biblioteca de recursos gráficos utilizado para el desarrollo de apps web con una adaptación para cada pantalla de los dispositivos permitiendo dar un contorno mas presentable en los distintos componentes como botones, paneles, etc.

Esta herramienta también fue implementada dentro de nuestro proyecto llegando a utilizar las diferentes propiedades que nos brinda este CSS implementado en diferentes puntos dentro de nuestro proyecto el cual nos brinda una mejor vista de nuestra página web para el usuario final.

Para la implementación de Bootstrap en nuestro proyecto se procede a ejecutar los siguientes comandos desde la terminal de Visual Studio Code.

- ***Instalación***

- ***npm install bootstrap --save***

Ejecutamos este comando dentro del terminal en la cual nos permite instalar la librería de iconos de Bootstrap para poder utilizar dentro de nuestro proyecto.

- ***Instalación***

- ***npm i bootstrap-icons --save***

- ***Angular Material***

Material Theming es la capacidad de personalizar sistemáticamente Material Design para reflejar mejor la marca de su producto. Cuando comienza a cambiar aspectos de su interfaz de usuario, como el color y la tipografía, las herramientas de Tematización de materiales aplican su visión de diseño a lo largo de su experiencia de usuario. Estas herramientas permiten cambiar fácilmente entre los flujos de trabajo de código y diseño al proporcionar valores específicos para todos los atributos personalizables. La personalización de estos valores crea un tema material para su producto (Angular, 2021).

Para la implementación dentro de nuestro proyecto debemos ejecutar el siguiente comando, el cual nos permitirá utilizar las diferentes componentes que nos ofrece esta herramienta.

Intalación: • ***ng add @angular/material***

3.5.5.0.2 Creación de Componentes

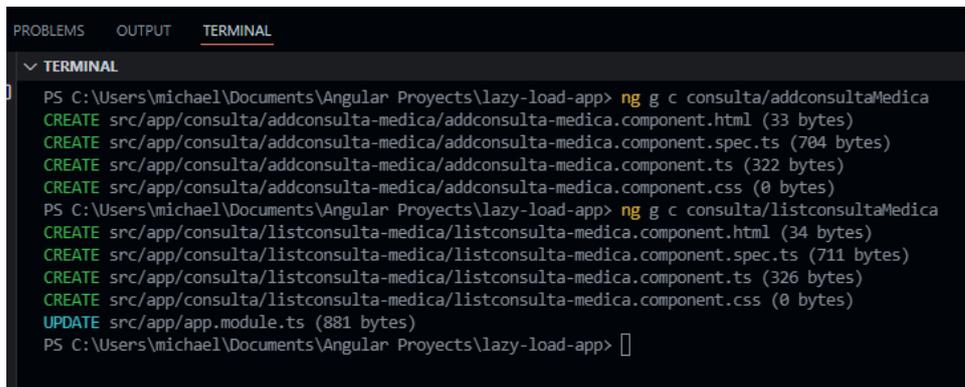
Primero nos situamos en la carpeta de nuestro proyecto la cual fue creado anteriormente con el nombre que se detalla a continuación.

- ***cd ./SistemaMedicoOcupacionaApp***

Ahora vamos a ejecutar el siguiente comando, el mismo que nos permite generar un componente con todos los archivos necesarios para la ejecución el mismo, para este ejemplo estamos generando un componente para agregar una consulta médica la mista que estará situado dentro de una carpeta consulta, de igual forma realizamos para el Listar.

- ***ng g c consulta/addconsultaMedica***

- ***ng g c consulta/listconsultaMedica***



```
PROBLEMS  OUTPUT  TERMINAL
└─ TERMINAL
  PS C:\Users\michael\Documents\Angular Proyectos\lazy-load-app> ng g c consulta/addconsultaMedica
  CREATE src/app/consulta/addconsulta-medica/addconsulta-medica.component.html (33 bytes)
  CREATE src/app/consulta/addconsulta-medica/addconsulta-medica.component.spec.ts (704 bytes)
  CREATE src/app/consulta/addconsulta-medica/addconsulta-medica.component.ts (322 bytes)
  CREATE src/app/consulta/addconsulta-medica/addconsulta-medica.component.css (0 bytes)
  PS C:\Users\michael\Documents\Angular Proyectos\lazy-load-app> ng g c consulta/listconsultaMedica
  CREATE src/app/consulta/listconsulta-medica/listconsulta-medica.component.html (34 bytes)
  CREATE src/app/consulta/listconsulta-medica/listconsulta-medica.component.spec.ts (711 bytes)
  CREATE src/app/consulta/listconsulta-medica/listconsulta-medica.component.ts (326 bytes)
  CREATE src/app/consulta/listconsulta-medica/listconsulta-medica.component.css (0 bytes)
  UPDATE src/app/app.module.ts (881 bytes)
  PS C:\Users\michael\Documents\Angular Proyectos\lazy-load-app> □
```

Figura 3.28: Generando compoente Angular.

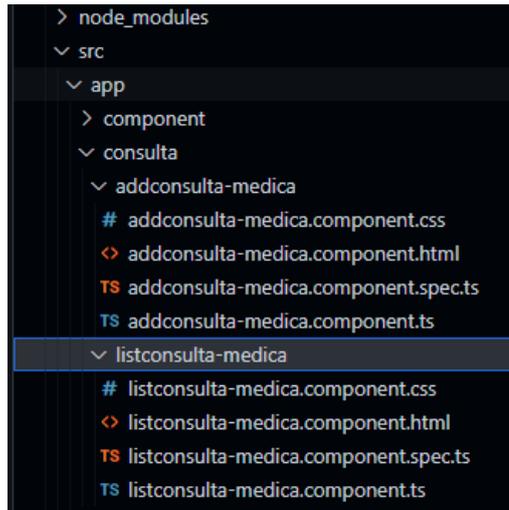


Figura 3.29: Componente Consulta Médica.

3.5.6. Implementación de Web Service Api REST

(Hat, 2020) “Las API son conjuntos de definiciones y protocolos que se utilizan para diseñar e integrar el software de las aplicaciones.” Por ejemplo, el diseño de una API de servicio meteorológico podría requerir que el usuario escribiera un código postal y que el productor diera una respuesta en dos partes: la primera sería la temperatura máxima y la segunda, la mínima.

Para la comunicación desde nuestro Front-end con el servidor Backend lo realizamos a través del API REST el mismo que permite interactuar con el servidor, logrando así obtener los datos que el Front-end está solicitando, para ello implementamos un módulo llamado HttpClient. Angular presenta un mecanismo llamado HttpClient el mismo que permite realizar peticiones REST para obtener una comunicación.

Implementación de HttpClient en Angular, importamos el componente después de ello le ingresamos dentro de los imports, todo ello lo realizamos dentro del fichero app.module.ts

Finalmente, creamos un componente de servicio para la implementación del API REST en el cual se lo realiza con la siguiente línea de código dentro del terminar ingresamos `ng g s service`, en este componente nos permitirá implementar los métodos de petición HTTP de los servicios

```
import { HttpClientModule } from '@angular/common/http';  
  
imports: [  
  HttpClientModule,  
]
```

Figura 3.30: Configuración del módulo HttpClient.

REST, como pueden ser lo que se detallan a continuación.

- ***Petición GET***

Este presente método mediante una petición, permite obtener cualquier información ya sea devolviéndome una lista o una entidad para luego ser mostrada mediante el Front-end hacia el usuario Final.

- ***Petición POST***

Este método es usado cuando se requiere enviar una información desde el cliente para luego en el servidor (Backend) persistir los datos y no se pierdan, es decir el usuario podría recuperar dicha información en algún momento si el así lo solicita.

- ***Código del servicio de Angular usando HttpClient***

3.5.6.0.1 Otras Utilidades

Se usaron estas librerías que son necesarias para agregar características adicionales como imprimir un reporte en PDF, mediante estas herramientas que se presentan a continuación podemos darle una estructura a un reporte que se implementó dentro de nuestro proyecto.

```

@Injectables({
  providedIn: 'root'
})
export class ServiceService {

  constructor(private http: HttpClient) { }

  //URL PACIENTE

  UrlList = 'SistemaMedicoOcupacional/srv/pacientes/listarPacientes';
  UrlInsert = 'SistemaMedicoOcupacional/srv/pacientes/regarPaciente';
  UrlListId = 'SistemaMedicoOcupacional/srv/pacientes/pacientesPorCodigo';
  UrlUpdate = 'SistemaMedicoOcupacional/srv/pacientes/actualizarPaciente';
  UrlDelete = 'SistemaMedicoOcupacional/srv/pacientes/eliminarPaciente';
  //UrlUpdate='SistemaMedicoOcupacional/srv/pacientes/actualizarPacienteId';

```

Figura 3.31: Web Service Angular.

```

// PACIENTES

getPacientes() {
  return this.http.get<Paciente[]>(this.UrlList);
}

createPaciente(paciente: Paciente) {
  return this.http.post<Paciente>(this.UrlInsert, paciente);
}

getPacienteId(codigo: number) {
  return this.http.get<Paciente>(this.UrlListId + "?codigo=" + codigo);
}

updatePaciente(paciente: Paciente) {
  console.log("prueba_002 ", paciente);
  //console.log("prueba_003 ", this.UrlUpdate+"?"+"codigo="+paciente.codigo, paciente);
  return this.http.put<Paciente>(this.UrlUpdate, paciente);
}

deletePaciente(paciente: Paciente) {
  return this.http.delete<Paciente>(this.UrlDelete + "?" + "codigo=" + paciente.codigo);
}

```

Figura 3.32: Web Service Angular.

- - “npm install pdfmake html-to-pdfmake jspdf –save”
- - “npm install –save pdfmake”
- - “npm install html-to-pdfmake”
- - “npm install jspdf –save”
- - “npm install html2canvas”
- - “npm install @types/jspdf @types/html2canvas -D”
- - “npm install –save @types/pdfmake”⁷⁷
- - “npm install –save @types/html-to-pdfmake”
- - “npm install jspdf jspdf-autotable”

3.5.7. Tecnología WAI-ARIA

La implementación de esta tecnología nos permite crear páginas web más accesibles al usuario, permitiendo al usuario obtener información relativamente importante de cómo está constituida la página, sobre todo aportando información, gracias al uso de Landmark Roles que permiten marcar adecuadamente el contenido.

También se debe complementar con el uso de etiquetas ARIA la cual nos permite agregar a los elementos ayudar accesible oh texto descriptivo brindando una mejor descripción de los elementos encontrados en la página web.

- Los landmark roles más usados para una página WEB son:
 - **Role Banner:** Este puede estar ubicado oh ser la cabecera de la pagina en la cual debe ser usado solo una vez.
 - **Role Navigation:** Usado para los elementos que forman un menú el cual puede ser empleado más de una vez en una página.
 - **Role Main:** Usado para identificar lo que respecta al contenido principal de la página.
 - **Role complementary:** Como su nombre lo indica sirve para marcar contenido no relevante dentro de la página, ya que solo es complementario de la misma.
 - **Role Search:** Usado principalmente para marcar información que se encuentre en el pie de la página en el cual solo debe ser empleado una sola vez.
 - **Role Contentinfo:** Para señalar la zona donde se encuentra un buscador dentro de la página en donde pueden existir más de una la vez.
 - **Role Form:** Como su nombre lo indica permite señalar la zona donde se encuentre un formulario, el cual este rol puede utilizar más de una vez.

3.5.7.0.1 Implementación WAI-ARIA en nuestro proyecto

- Porque es necesario ARIA
 - La mayoría de aplicaciones utilizan diferentes tipos de widgets, como modales, pestañas en la página, menús desplegables de botones, selectores de fechas entre otros. Estos widgets a menudo incluyen interacciones dinámicas, como la capacidad de seleccionar, verificar, arrastrar y soltar y cambiar el tamaño, ocultar y mostrar e incluidos muchos más. A menudo se denominan no nativos porque se basan en JavaScript permitiendo un control de las interacciones en lugar de utilizar HTML nativo, debido que se basan en JavaScript estos widgets son problemáticos para las tecnologías de asistencia y por lo tanto son inaccesibles para las personas con discapacidad. Por Ejemplo, un ComboBox no saben el estado del widget, es decir si está abierto o cerrado, con la tecnología ARIA se utiliza para resolver este tipo de inconveniente.

- Funcionamiento de ARIA
 - Esta tecnología permite a los desarrolladores web aplicar atributos HTML determinado a elementos HTML, los mismos que no modifican la experiencia ni el comportamiento de ningún elemento HTML, sin embargo, realizan lo siguiente orientado a tecnologías de asistencia.
 - Primero: Agregan un significado adicional a los elementos HTML.
 - Dos: Pueden modificar si el significado existe de los elementos HTML.
 - Tres: Pueden agregar etiquetas o descripciones adicionales a los elementos HTML.
 - Cuarto: Pueden establecer relaciones entre elementos HTML.
 - Cinco: Pueden informar a las tecnologías de asistencia que se han producido modificaciones en áreas específicas de una página HTML.

- Roles, Estados y Propiedades.

- En realidad, ARIA es una serie de atributos HTML, las mismas que se clasifican como funciones, estados o propiedades.
- Los atributos de Rol permiten a cada uno de los desarrolladores informar a las tecnologías de asistencia que tipo de widget es.
- Por ejemplo, un widget, una barra de progreso, etc.
- Dentro de nuestro proyecto se definió un widget en la cual utilizamos un Rol de menú, como se puede apreciar en la imagen.

```
<ul role="menu" style="margin-top: 3%;">
  <li *ngFor="let diagnostico of diagnosticoSelect" class="text-center">
    <h3> Diagnostico seleccionado </h3>
    <h2> {{diagnostico.nombre}} </h2>
  </li>
</ul>
```

Figura 3.33: Aplicando Aria Role en proyecto.

Los atributos de estado nos permiten informar a las tecnologías de asistencia cual es el estado actual del widget. Veamos la implementación este atributo de estado en nuestro proyecto, la cual se muestra en la siguiente imagen.

En la siguiente figura el widget que este marcado, deshabilitado o en alguno otro estado.

```
<mat-grid-tile>
  <div class="form-group">
    <mat-label id="example-radio-group-label">Sexo</mat-label>
  </div>
  <div aria-checked="true" class="form-group">
    <mat-radio-group aria-labelledby="example-radio-group-label" class="example-radio-group"
      [(ngModel)]="paciente.sexo">
      <mat-radio-button class="example-radio-button" *ngFor="let season of seasons"
        [value]="season"
        {{season}}
      </mat-radio-button>
    </mat-radio-group>
  </div>
</mat-grid-tile>
```

Figura 3.34: Tecnología ARIA.

En este ejemplo el primer elemento tiene un atributo que informa a la tecnología de asisten-

cia que el elemento está en un estado marcado.

Los atributos de propiedad nos permiten informar a las tecnologías de asistencia el propósito del elemento o si tiene una relación con otro elemento. También como podemos apreciar en la imagen estamos utilizando el atributo `aria-labelledby`, estable una relación entre el elemento de entrada y el contenido dentro el elemento.

```
<div style="margin-bottom: 1%;" class="card-footer bg-transparent border-success" role="contentinfo"
  aria-labelledby="Guardar Paciente y Retorno a la lista de pacientes">
  <div mat-dialog-actions align="center">
    <button (click)="guardar()" mat-button mat-raised-button color="primary">Guardar Paciente</button>
  </div>
</div>
```

Figura 3.35: Tecnología ARIA.

De igual forma presentamos una imagen de nuestro proyecto, en este caso aplicando `aria-label` ingresado dentro de un botón el mismo que presenta una información adicional solo para la tecnología de asistencia.

Utilizado un `Role = 'Contentinfo'`, como se describía anteriormente, mediante este rol definimos nuestro pie de pagina en la cual identifica los datos en la parte inferior de cada una de las paginas.

3.5.7.0.2 Soporte o apoyo de tecnología WAI-ARIA

- Existe una amplia gama de navegadores los mismos que la gran mayoría de los mismos son compatibles con ARIA incluido:
 - Firefox 3 y superior.
 - Internet Explorer 8 o superior.
 - Safari OSX 5 o superior.
 - Chrome 7 o superior.

También, ARIA es compatible con un sin numero de tecnologías de asistencia, incluido JAWS 8 y superior, ZoomTexto, VoiceOver para Mac OS e IOS; NVDA y ORCA, teniendo en cuenta que el apoyo es un término genera, cada combinación de navegador y tecnología de asistencia tiene sus propias peculiaridades e inconsistencias menores, además la especificación ARIA está en constante evolución

Capítulo IV

RESULTADOS

4.1. Módulos del Sistema Web

Cada uno de los módulos o interfaces de usuario cuentan con distintas acciones u operaciones básicas CRUD (Crear, Leer, Actualizar, Eliminar) también acciones como interfaces para Login de Usuarios según roles establecidos, registrar exámenes de Laboratorio, Espirometría, EKG, Ver Exámenes, Audiometría, Radiología, Reportes entre otras más relacionadas a la Medicina Ocupacional.

- El sistema contará con distintos módulos que estarán disponibles de acuerdo con el tipo de usuario autenticado.
- El sistema deberá permitir registrar una nueva Consulta (Médica, Preventiva, Preocupacional) de acuerdo con el módulo seleccionado.
- El sistema debe tener registrado los pacientes/trabajadores pertenecientes a la Empresa, los cuales deben ser cargados previamente al iniciar el módulo que el usuario seleccione.
- El sistema deberá llevar el historial médico de cada uno de los pacientes.
- El sistema deberá permitir generar Certificados Médicos.

- El sistema debe tener registrado previamente todos los capítulos referentes a las distintas enfermedades existentes en la medicina los cuales deben ser cargados y visualizados cuando el usuario lo solicite.
- El sistema debe tener registrado previamente todos los Diagnósticos pertenecientes a cada Síndrome/Enfermedad existentes en la medicina los cuales deben ser cargados y visualizados cuando el usuario lo solicite.
- Generar Reportes por cada uno de los módulos implementados indicando Fecha Inicio y Fecha Final.

4.1.1. LOGIN

Este es el primer módulo que se presentará en la página web donde el sistema deberá permitir la autenticación de usuarios para el ingreso a la aplicación y a todas sus características, en el cual por defecto se ha configurado un usuario desde el Sistema Gestor de BD llamado administrador que tiene acceso a todos los modulos implementados en la página web. El cual podrá agregar un nuevo usuario entregándole su “username” y “password”.

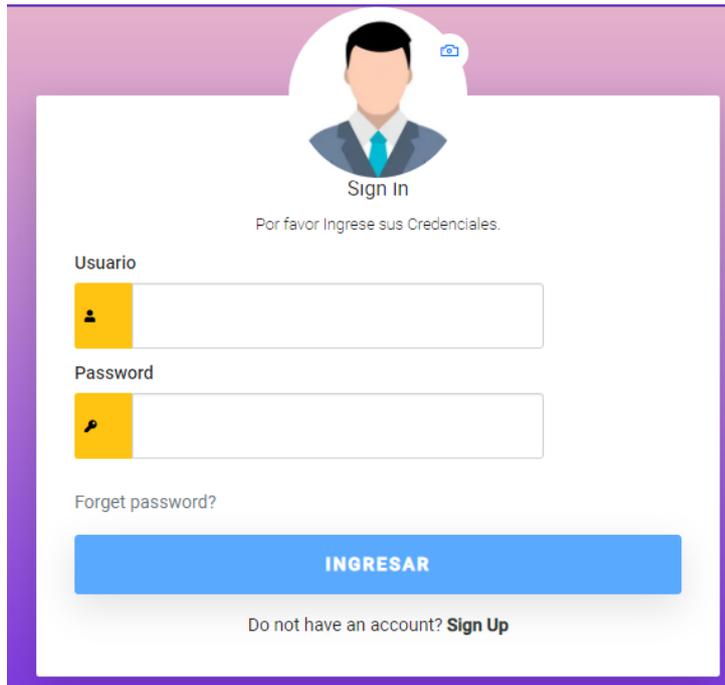


Figura 4.1: Autenticación Usuarios

Al ingresar si la información proporcionada son invalidas obtendrá un notificación de error relacionado a los campos que haya llenado incorrectamente llamado “Usuario o password incorrectos”, caso contrario si las credenciales son correctas podrá usar visualizar y hacer uso de todos los módulos presentados en el menú.

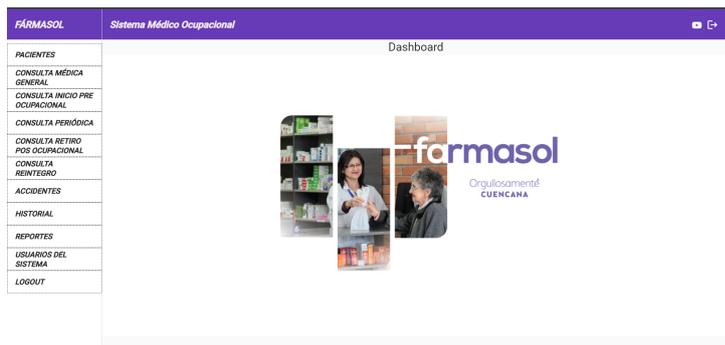


Figura 4.2: Menú Principal

Para ello se usó un servicio Rest que devuelve "true"ó "false"dependiendo si el usuario existe o no en el sistema para el respectivo acceso al sistema.

Login

Parámetros de entrada

Username: Nombre de usuario asignado para ingresar al sistema.

Password: Clave para ingresar al sistema.

```
@GET
@Path("/login")
@Produces("application/json")
public String Login(@QueryParam("username") String username,@QueryParam("password") String password){
    return uon.Login(username, password);
}
```

Figura 4.3: Método Login

Ademas se usó un servicio Rest para obtener el nombre de usuario que accede al sistema para hacer uso mas adelante del mismo en cada uno de los módulos.

Get Usuario Por Username

Parámetros de entrada

Username: Nombre de usuario registrado en el sistema.

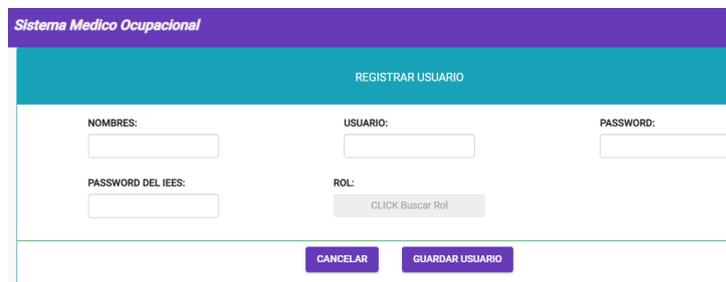
```
@GET
@Path("/getUsuarioLogin")
@Produces("application/json")
public Usuario getUsuariosPorUsername(@QueryParam("username") String username){
    return uon.getUsuarioPorUsername(username);
}
```

Figura 4.4: Método Obtener Usuario por Username

4.1.2. MÓDULO USUARIOS DEL SISTEMA

Este módulo permitirá al usuario administrador poder gestionar todos los usuarios registrados en el sistema los cuales serán mostrados en un listado. A su vez, el administrador podrá ya sea registrar un nuevo usuario brinda toda la información correcta para la creación el mismo como también asignarle un rol de acuerdo a las necesidades (Médico o Enfermero) y así finalmente pueda tener acceso a las funcionalidades del sistema.

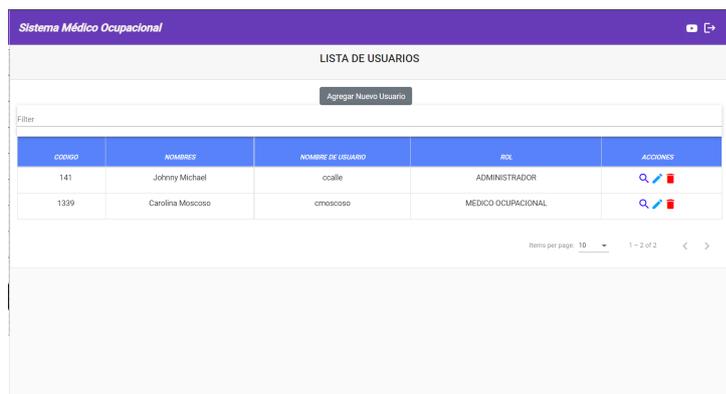
Si todo fue correcto se mostrará un mensaje indicando “Datos guardados exitosamente”.



The screenshot shows a web form titled "REGISTRAR USUARIO" within a purple header "Sistema Médico Ocupacional". The form has four input fields: "NOMBRES:", "USUARIO:", "PASSWORD:", and "PASSWORD DEL IEES:". Below the "USUARIO:" field is a button labeled "CLICK Buscar Rol". At the bottom of the form are two buttons: "CANCELAR" and "GUARDAR USUARIO".

Figura 4.5: Registrar Usuario

Además, el administrador podrá modificar toda la información almacenada anteriormente en caso de haber proporcionado algún dato erróneo eligiendo el usuario a ser modificado de la lista desplegada.



The screenshot shows a web page titled "LISTA DE USUARIOS" within a purple header "Sistema Médico Ocupacional". At the top, there is a button "Agregar Nuevo Usuario" and a "Filter" input field. Below is a table with the following data:

CODIGO	NOMBRE	NOMBRE DE USUARIO	ROL	ACCIONES
141	Johnny Michael	ccalle	ADMINISTRADOR	[Iconos de acción]
1399	Carolina Moscoso	cmoscoso	MEDICO OCUPACIONAL	[Iconos de acción]

At the bottom right of the table area, there is a pagination control: "Items per page: 10" and "1 - 2 of 2".

Figura 4.6: Lista de Usuarios

4.1.3. MÓDULO MENÚ

En este módulo se presentará una lista de acciones los cuales podrán ser redireccionados a las distintas funcionalidades con el cual cuenta el sistema.

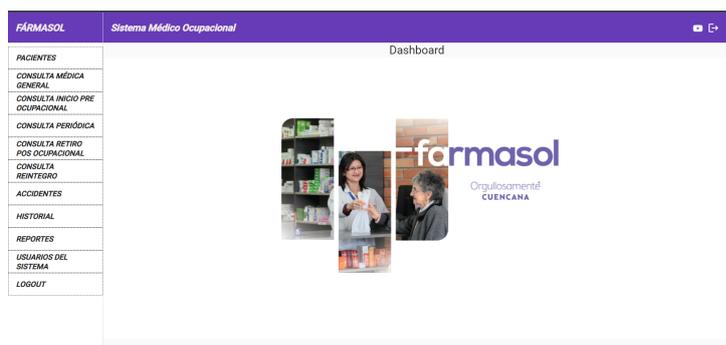
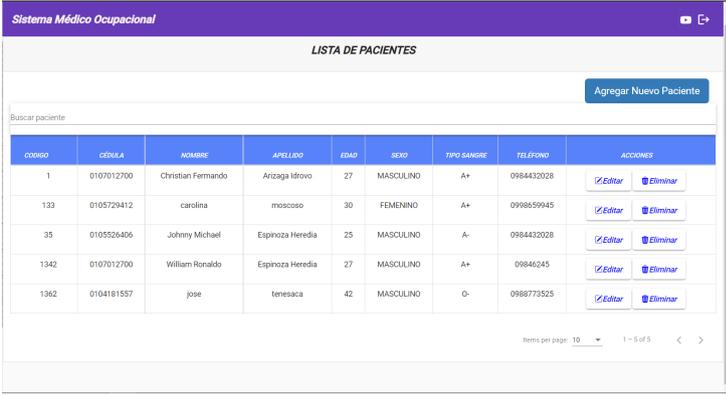


Figura 4.7: Menú Principal

4.1.4. MÓDULO PACIENTES

Este módulo mostrará una lista de todos los trabajadores o empleados pertenecientes a la empresa los cuales a su vez pueden ser filtrados ya sea por nombre, cedula, código o apellido haciendo más ágil y optimizando el tiempo de búsqueda.



CODIGO	CEDULA	NOMBRE	APELLIDO	EDAD	SEXO	TIPO SANGRE	TELEFONO	ACCIONES
1	0107012700	Christian Fernando	Arizaga Idrovo	27	MASCULINO	A+	0984432028	Editar Eliminar
133	0105729412	carolina	moscoso	30	FEMENINO	A+	0998559945	Editar Eliminar
35	0105526406	Johnny Michael	Espinoza Heredia	25	MASCULINO	A-	0984432028	Editar Eliminar
1342	0107012700	William Ronaldo	Espinoza Heredia	27	MASCULINO	A+	098460245	Editar Eliminar
1362	0104181557	jose	tenesaca	42	MASCULINO	O-	0988773525	Editar Eliminar

Figura 4.8: Lista de Pacientes

Dentro de este podremos agregar o registrar un nuevo paciente proporcionando toda la información relevante para su creación, además cuenta con un método que permite calcular la edad del paciente recibiendo como parámetro la fecha de nacimiento.

Se podrá actualizar la información previamente almacenada pero no se lo podrá eliminar ya que debido a los requisitos el paciente o trabajador solo podrá ser dado de baja actualizando el parámetro específico **“ACTIVO O INACTIVO”** ya que es necesario mantenerlo en el registro.

Luego al finalizar sobre todo el registro ya estará habilitado para poder realizar sobre este las distintas acciones relacionadas a la Medicina Ocupacional.

REGISTRAR PACIENTE

Cédula:	Cargo:	<input type="button" value="CREAR CARGO"/>	Nombre:	Apellido:
<input type="text"/>	<input type="text"/>		<input type="text"/>	<input type="text"/>
Domicilio:	Teléfono:	Fecha de Nacimiento:	Edad:	Fecha de Ingreso Empresa:
<input type="text"/>	<input type="text"/>	<input type="text" value="DD/MM/AAAA"/>	<input type="text"/>	<input type="text" value="28/03/2022"/>
SEXO:	Tipo de Sangre:	Estado Civil:	Instrucción:	Ciudad:
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Provincia:	Nombre de Contacto:	Teléfono de Contacto:	Parentesco:	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	

Figura 4.9: Registrar Paciente

4.1.5. MÓDULO CONSULTA MEDICA

Uno de los módulos principales del sistema es el de consulta médica ya que es uno de los que más uso tendrá permitiendo registrar una nueva consulta médica ingresando los parámetros en el sistema obtenidos previamente del paciente, a su vez si un paciente ya tuvo una consulta anteriormente el médico tendrá acceso a esa información ya que podrá visualizar en una tabla o listado la información más relevante, teniendo conocimiento de forma general el diagnóstico del paciente.

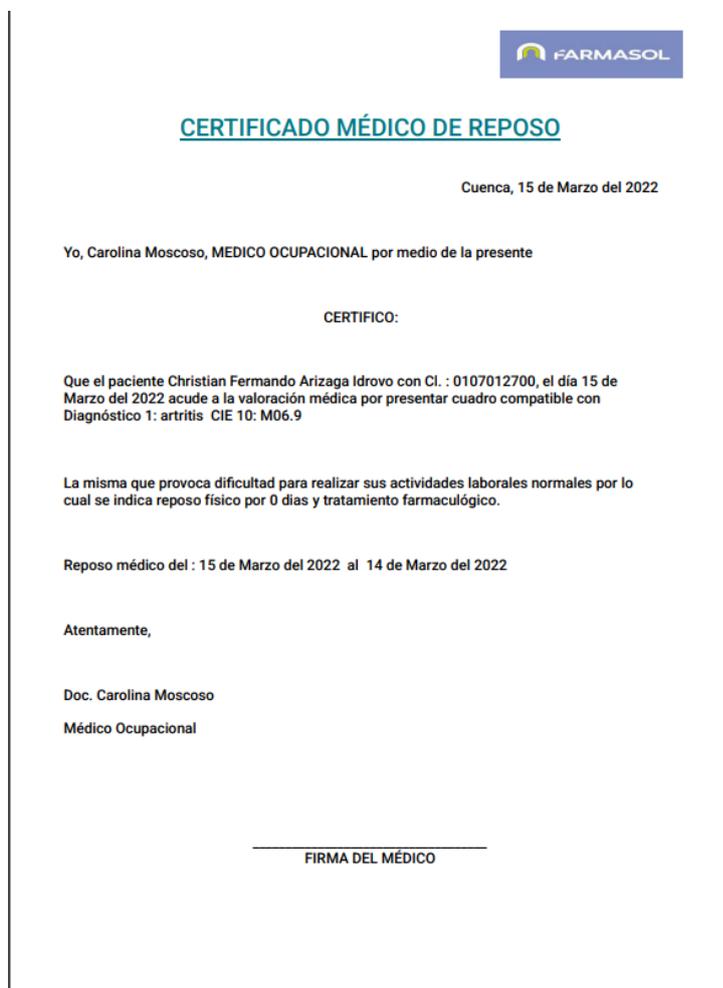
Figura 4.10: Registrar Consulta Médica

CÓDIGO	FECHA	PESO	TALLA	MASA CORPORAL	ESTADO NUTRICIONAL	TEMPERATURA	MOTIVO CONSULTA	PROBLEMA ACTUAL	ACCIONES
1239	2022-03-15	70	1.65	25.71	SOBREPESO	78			Editar Eliminar Certificado Reposo
1244	2022-03-15	80	1.86	23.12	PESO NORMAL	43			Editar Eliminar Certificado Reposo
1290	2022-03-20	0	0	0		0			Editar Eliminar Certificado Reposo
1298	2022-03-20	0	0	0		0			Editar Eliminar Certificado Reposo
1363	2022-03-28	0	0	0		0			Editar Eliminar Certificado Reposo

Figura 4.11: Lista Consultas Médicas

En el listado presentado el usuario tendrá la opción de búsqueda de una consulta en específico como también acciones sobre la misma como son editarla o eliminarla según así lo requiera.

Una de las funcionalidades importantes dentro de este módulo es el poder generar un certificado médico de reposo en formato PDF útil para pacientes que requieran de descanso, optimizando así el tiempo que el médico emplearía en realizarlo manualmente. Para ello simplemente al dar clic generar certificado ubicado en la parte derecha del listado se abrirá el PDF con la información requerida como nombre del médico, nombre del paciente, diagnóstico, fecha de consulta y un apartado que será firmado por el médico responsable.



The image shows a PDF document titled "CERTIFICADO MÉDICO DE REPOSO" from FARMASOL. The document is dated "Cuenca, 15 de Marzo del 2022". It is issued by "Yo, Carolina Moscoso, MEDICO OCUPACIONAL por medio de la presente". The patient is "Christian Fernando Arizaga Idrovo" with CI: 0107012700, who presented on March 15, 2022, with a diagnosis of "artritis CIE 10: M06.9". The certificate states that the patient has difficulty performing normal work activities, and physical rest for 0 days and pharmacological treatment are indicated. The medical rest is from March 15, 2022, to March 14, 2022. The doctor's name is "Doc. Carolina Moscoso, Médico Ocupacional". There is a line for the doctor's signature labeled "FIRMA DEL MÉDICO".

Figura 4.12: Certificado Médico

Para ello se usó la herramienta pdfmake de angular la cual obtiene los datos de un servicio Rest que devuelve la Consulta de un paciente.

Get Consulta por Código

Parámetros de entrada

Id:Código de la Consulta.

```
@GET
@Path("/consultaMedicaPorCodigo")
@Produces("application/json;charset=utf-8")
public ConsultaMedica getConsultaPorCodigo(@QueryParam("codigo")Integer id) {
    return con.getConsultaMedicaPorCodigo(id);
}
```

Figura 4.13: Método obtener datos de consulta

```
async createPdfCertificado() {
    const pdfDefinition: any = {
        content: [
            {
                image: await this.getBase64ImageFromURL(
                    "https://encrypted-tbn0.gstatic.com/images?q=tbn:AND9GcQUAawmgW5VAGNUJobGg-tA6eEd8eWnQeMhJgxt"
                ),
                fit: [150, 150],
                alignment: 'right',
                // width: 150,
                // height: 150,
            },
            {
                text: 'CERTIFICADO MÉDICO DE REPOSO',
                decoration: 'underline',
                bold: true,
                fontSize: 20,
                alignment: 'center',
                color: '#047886',
                margin: [0, 30, 0, 20]
            },
        ],
        columns: [
            [
                {
                    text: 'Cuenca, ' + this.fechaDiaMesAnio,
                    bold: true,
                    alignment: 'right',
                    margin: [5, 10, 20, 30]
                },
                {
                    text: 'Yo, ' + this.nombreUsuario + ", " + this.rolUsuario + " por medio de la presente",
                }
            ]
        ]
    };
}
```

Figura 4.14: PDF MAKE

4.1.6. MÓDULO CONSULTA INICIO/PREOCUPACIONAL

El sistema permitirá realizar consultas de exámenes médicos preocupaciones, teniendo como intención de determinar la aptitud del paciente acorde a cada una de sus condiciones psicofísicas o detectar patologías preexistentes para el ocupación de las actividades que tendran designadas para cada uno de ellos.

Este módulo permitirá registrar una consulta de inicio preocupacional la cual es indispensable para mantener el control de todas las consultas que cada empleado en la empresa tiene que realizarse cuando inicia sus labores por primera vez, ingresando así todos los parámetros requeridos.

The screenshot shows a web form titled "REGISTRAR CONSULTA PREOCUPACIONAL". At the top, there are four input fields: "Cédula:" with value "0107012700", "Paciente:" with value "Christian Fernando Arizaga Idrovo", "Fecha Ingreso:" with value "09/03/2022", and "Cargo Paciente:" with value "CONTADOR". Below this is a navigation bar with three tabs: "PREOCUPACIONAL INICIO 1-3" (selected), "PREOCUPACIONAL INICIO 2-3", and "PREOCUPACIONAL INICIO 3-3". The main form is divided into two sections: "A. DESCRIPCIÓN" and "B. ANTECEDENTES GINECO OBSTÉTRICOS". Section A includes "MOTIVO DE CONSULTA:" (Description) and "ANTECEDENTES PERSONALES:" (FERNANDO... PERIODICA). Section B includes fields for "Menarquia:", "Ciclos:", "F.U.M." (Fecha O/I/M/A), "Gestas:", "Partos:", "Cesáreas:", "Abortos:", "Vida Sexual Activa" (dropdown), "PAPANICOLAOU" (dropdown), "Tiempo (Años):" (Años), and "Resultado:" (Description).

Figura 4.15: Registrar Consulta Inicio/Preocupacional

A la vez luego de haber realizado el proceso anterior se podrá observar mediante un listado todas las consultas realizadas en específico las cuales dependiendo de la necesidad podrán ser actualizadas o eliminadas según la acción requerida por el médico ocupacional.

Paciente: 0105729412 carolina moscoso • LISTA CONSULTAS PREOCUPACIONAL INICIAL

VOLVER ATRÁS NUEVA CONSULTA PREOCUPACIONAL

Buscar

CODIGO	FECHA	PESO	TALLA	MAESA CORPORAL	ESTADO NUTRICIONAL	ENFERMEDAD ACTUAL	TRATAMIENTO	CERTIFICADO	ACCIONES
142	2022-03-11	70	1.65	25.71	PESO NORMAL	BHJJHJH	BHJJHJHJ	Certificado Aptitud	<input type="checkbox"/> Visualizar <input type="checkbox"/> Eliminar <input type="checkbox"/> Imprimir Consulta
159	2022-03-12	70	1.12	55.8	OBESIDAD	NIGNUNA	ok	Certificado Aptitud	<input type="checkbox"/> Visualizar <input type="checkbox"/> Eliminar <input type="checkbox"/> Imprimir Consulta

Figura 4.16: Lista Consultas Inicio/Preocupacional

El módulo tendrá el apartado para registrar datos de exámenes realizados previamente a un Paciente cómo:

- Exámenes de laboratorio (Hemograma, Químico, Serología, Orina, Heces).
- Registrar resultados de Electrocardiograma.
- Registrar Resultados de Radiología.
- Registrar resultados de Espirometría: Este apartado cuenta con la opción de realizar el cálculo de la Capacidad Vital Forzada (FVC), Volumen Espiratorio Forzado en el primer segundo (FEV1) y el cálculo de la Relación FEV1/FVC según los parámetros necesarios para cada una de las acciones permitiendo así observar el resultado de cada uno de ellos en base a las operaciones realizadas.
- Registrar resultados Audiometría: En este apartado se realiza el cálculo de cada uno de los índices correspondientes a un examen de audiometría como son el índice ELI, índice SAL, PGA(AAO) Y PGA(AMA) para posteriormente mostrar y observar cada uno de los resultados basados en las distintas operaciones mencionadas y así permitir determinar el estado auditivo de un paciente.

Fecha: 04/01/2022 EXAMEN ESPIROMETRIA

MED	PRED	PRED%
FVC		
FEV1		
FEV1/FVC		
Resultado FVC:	Resultado FEV1:	Resultado:

OBSERVACIONES: Observaciones..

Cancelar Confirmar

Figura 4.17: Registro Exámen Espirometria

Fecha: 04/01/2022 EXAMEN AUDIOMETRIA

	125	250	500	1000	2000	3000	4000	6000	8000
OIDO IZQUIERDO									
OIDO DERECHO									

Tiempo de Exposición (Horas)

dB Exposición Resultado

ÍNDICE S.A.L. CALCULAR INDICES

Oido Izquierdo Resultado

Cancelar Confirmar

Figura 4.18: Registro Exámen Audiometría

Fecha: 04/01/2022 ¡EXAMEN LABORATORIO!

Hemograma Quimico Serologia Orina Heces

Globulos Rojos:	Hemoglobina:	Hematocrito:	Leucocitos:
Neutrofilos:	Linfocitos:	Eosinofilos:	Basofilos:
Monositos:	VSG:	Plaquetas:	Estado

OBSERVACIONES: Observaciones..

Figura 4.19: Registro Exámen Laboratorio

EXAMEN ELECTROCARDIOGRMA

Fecha: 04/01/2022 Resultado

OBSERVACIONES: Observaciones..

Cancelar Confirmar

Figura 4.20: Registrar Electrocardiograma

Fecha: 04/01/2022 EXAMEN RADIOLOGIA

RX Tórax RX Columna

OBSERVACIONES: Observaciones..

Cancelar Confirmar

Figura 4.21: Registrar Radiología

Finalmente, el módulo tiene la opción de imprimir en formato PDF todos los datos correspondientes a una Consulta Inicio PreOcupacional que el médico haya seleccionado para su respectiva visualización.

4/1/22, 12:47 AM CONSULTA MÉDICA OCUPACIONAL / INICIO

DATOS DEL PACIENTE

Empresa:	Cédula:	Paciente:	Género:	Cargo Paciente:
FARMASOLEP	0105729412	carolina moscoso	FEMENINO	CONTADOR

A. DESCRIPCIÓN

MOTIVO DE CONSULTA:	gubhnykm	ANTECEDENTES PERSONALES:	kkkkkkkkkk NUEVO2
---------------------	----------	--------------------------	-------------------

B. ANTECEDENTES GINECO OBSTÉTRICOS

Menarquia:	0	Ciclos:	0	F.U.M.:		Gestas:	0	Partos:	0	Cesáreas:	0
Abortos:	0	Vida Sexual Activa ▼									
PAPANICOLAOU ▼	Tiempo (Años):		Resultado:								
ECO MAMARIO ▼	Tiempo (Años):		Resultado:								
COLPOSCOPIA ▼	Tiempo (Años):		Resultado:								
MAMOGRAFIA ▼	Tiempo (Años):		Resultado:								

Figura 4.22: Consulta Inicio/Preocupacional

4.1.7. MÓDULO CONSULTA PERIÓDICA

Este módulo permitirá generar una consulta médica preventiva, en la cual un médico ingresará al sistema los parámetros requeridos obtenidos previamente del paciente, esto indispensable ya que gracias al mantener el registro de todas las consultas preventivas se puede prevenir y evitar enfermedades que pueden llegar a convertirse sumamente serias en un paciente.

REGISTRAR CONSULTA PERIÓDICA

Cédula:	Paciente:	Fecha Ingreso:	Cargo Paciente:
0107012700	Christian Fernando Arizaga Idrovo	09/03/2022	CONTADOR

PERIÓDICA 1-2 PERIÓDICA 2-2 PERIÓDICA 3-3

DESCRIPCIÓN

MOTIVO DE CONSULTA:	Descripción:	ANTECEDENTES PERSONALES:	FERNANDO00 (PERIÓDICA)
---------------------	--------------	--------------------------	------------------------

E. HABITOS TÓXICOS

	Tiempo consumo (meses)	Cantidad	Ex consumidor	Tiempo de Abstinencia (meses)
TABACO: ▼	<input type="text"/>	<input type="text" value="0"/>	<input type="text"/>	<input type="text"/>
ALCOHOL: SI ▼	<input type="text"/>	<input type="text" value="0"/>	<input type="text"/>	<input type="text"/>
OTRAS DROGAS ▼	<input type="text"/>	<input type="text" value="0"/>	<input type="text"/>	<input type="text"/>

Figura 4.23: Registrar Consulta Periódica

Luego de haber realizado el proceso de registro el sistema mostrará mediante un listado todas las consultas realizadas y así según la necesidad que requiera el médico se las podrá actualizar o eliminar.

Paciente: 0107012700 Christian Fernando Arizaga Idrovo • LISTA DE CONSULTAS PERIÓDICAS

VOLVER ATRÁS NIEVA CONSULTA PERIODICA

Buscar

CODIGO	FECHA	PESO	TALLA	MASA CORPORAL	ESTADO NUTRICIONAL	ENFERMEDAD ACTUAL	TRATAMIENTO	CERTIFICADO	ACCIONES
1303	2022-09-20	60	1.93	16.11	PESO BAJO	PERIODICA	PERIODICAtratamiento	Certificado Aptitud	<input checked="" type="checkbox"/> Visualizar <input type="checkbox"/> Eliminar <input type="button" value="Imprimir Consulta"/>

Items per page: 10 0 of 0

Figura 4.24: Lista Consultas Periódica

Finalmente, el módulo tiene la opción de imprimir en formato PDF todos los datos correspondientes a una Consulta Periódica que el médico haya seleccionado para su respectiva visualización.

4/1/22, 12:51 AM CONSULTA MEDICA PERIODICA

DATOS DEL PACIENTE

Empresa: FARMASOL EP
Cédula: 0107012700
Paciente: Christian Fernando Arizaga I
Género: MASCULINO
Cargo Paciente: CONTADOR

DESCRIPCIÓN

MOTIVO DE CONSULTA: PERIODICA
ANTECEDENTES PERSONALES: FERNANDO I PERIODICA

E. HáBITOS TÓXICOS

	Tiempo consumo (meses)	Cantidad	Ex consumidor	Tiempo de Abstinencia (meses)
TABACO:	<input type="text"/>	<input type="text" value="0"/>	<input type="text"/>	<input type="text"/>
ALCOHOL:	<input type="text"/>	<input type="text" value="0"/>	<input type="text"/>	<input type="text"/>
OTRAS DROGAS	<input type="text"/>	<input type="text" value="0"/>	<input type="text"/>	<input type="text"/>

F. ESTILO DE VIDA

ACTIVIDAD FÍSICA: ¿CUÁL?: Descripción... **TIEMPO / CANTIDAD:**

MEDICACIÓN HABITUAL: ¿CUÁL?: Descripción... **TIEMPO / CANTIDAD:**

Figura 4.25: Consulta Periódica

4.1.8. MÓDULO CONSULTA RETIRO/POSOCUPACIONAL

Este módulo permite el registro de una consulta retiro pos-ocupacional la cual se la realiza cuando un trabajador ha culminado sus labores dentro de la empresa y se requiere realizar este tipo de consulta para mantener el registro del estado de salud al momento del retiro ingresando los parámetros requeridos que serán obtenidos del paciente.

REGISTRAR CONSULTA MEDICA DE RETIRO

Paciente: Christian Fernando Ariza Fecha de Ingreso: 09/03/2022 Cargo Paciente: CONTADOR Fecha Inicio Labores: Fecha de Salida: Tiempo:

RETIRO 1-2 RETIRO 2-2

A. FACTORES DE RIESGO

PUESTO DE TRABAJO: Descripción:

ACTIVIDADES: Descripción:

FISICO

ANTECEDENTES PERSONALES

J. ANTECEDENTES CLÍNICOS Y QUIRÚRGICOS

DESCRIPCIÓN: FERNANDO O O I P PERIODICA

Figura 4.26: Registrar Consulta Retiro/Posocupacional

Al finalizar el proceso anterior será redirigido hacia el listado de consultas donde de igual manera observará los datos mostrados y podrá realizar tanto la actualización de los datos si así lo requiere como también podrá eliminar la consulta del registro.

Paciente: 0107012700 Christian Fernando Arizaga Idrovo • LISTA CONSULTAS RETIRO

VOLVER ATRÁS NUEVA CONSULTA RETIRO

Buscar

CODIGO	FECHA	PESO	TALLA	MASA CORPORAL	ESTADO NUTRICIONAL	ENFERMEDAD ACTUAL	TRATAMIENTO	CERTIFICADO	ACCIONES
1327	2022-03-21	70	1.65	25.71	PESO NORMAL		tomar muchas pastillas	Certificado Aptitud	Visualizar Eliminar Imprimir Consulta

Figura 4.27: Lista Consultas Retiro/Posocupacional

Finalmente, el módulo tiene la opción de imprimir en formato PDF todos los datos correspondientes a una Consulta Retiro PosOcupacional que el médico haya seleccionado para su respectiva visualización.

4/1/22, 12:47 AM CONSULTA MÉDICA OCUPACIONAL / INICIO

DATOS DEL PACIENTE

Empresa:	Cédula:	Paciente:	Género:	Cargo Paciente:
FARMASOLEP	0105729412	carolina moscoso	FEMENINO	CONTADOR

A. DESCRIPCIÓN

MOTIVO DE CONSULTA:	gvubhnrjkm	ANTECEDENTES PERSONALES:	kkkkkkkkkk NUEVO2
---------------------	------------	--------------------------	-------------------

B. ANTECEDENTES GINECO OBSTÉTRICOS

Menarquia:	0	Ciclos:	0	F.U.M:		Gestas:	0	Partos:	0	Cesáreas:	0
Abortos:	0	Vida Sexual Activa ▼									
PAPANICOLAOU ▼	Tiempo (Años):		Resultado:								
ECO MAMARIO ▼	Tiempo (Años):		Resultado:								
COLPOSCOPIA ▼	Tiempo (Años):		Resultado:								
MAMOGRAFIA ▼	Tiempo (Años):		Resultado:								

Figura 4.28: Consulta Retiro/Posocupacional

4.1.9. MÓDULO CONSULTA REINTEGRO

Este módulo permite el registro de una consulta de reintegro la cual se lo realiza a todos los trabajadores que vuelven de un periodo de reposo y se incorporan nuevamente a su ámbito laboral por lo que es indispensable saber el estado de salud en el cuál se encuentren para ello el médico ingresará todos los datos requeridos para que se pueda guardar o registrar correctamente una consulta, de igual forma el módulo permite el cálculo automático del Índice de Masa Corporal como el estado nutricional del paciente según los parámetros requeridos.

The form is titled "REGISTRAR CONSULTA REINTEGRO" and contains the following fields:

- PACIENTE:** Christian Fernando Ariza
- Cargo Paciente:** CONTADOR
- Causa de salida:** [Empty text box]
- Fecha Última Día Laboral:** [Empty date box]
- Fecha de Reingreso:** [Empty date box]
- Tiempo:** [Empty text box]

A. MOTIVO DE CONSULTA / CONDICIÓN DE REINTEGRO

DESCRIPCIÓN: [Empty text box]

B. ENFERMEDAD ACTUAL

DESCRIPCIÓN: [Empty text box]

C. CONSTANTES VITALES Y ANTROPOMETRÍA

PRESIÓN ARTERIAL: [Empty text box]

TEMPERATURA: [Empty text box]

FRECUENCIA CARDIACA: [Empty text box]

SATURACIÓN OXÍGENO: [Empty text box]

FRECUENCIA RESPIRATORIA: [Empty text box]

Figura 4.29: Registrar Consulta Reintegro

Paciente: 0107012700 Christian Fernando Arizaga Idrovo • LISTA CONSULTAS REINTEGRO

VOLVER ATRÁS NUEVA CONSULTA REINTEGRO

Buscar

CODIGO	FECHA	PESO	TALLA	MASA CORPORAL	ESTADO NUTRICIONAL	ENFERMEDAD ACTUAL	TRATAMIENTO	CERTIFICADO	ACCIONES
1336	2022-09-21	70	1.65	25.71	PESO NORMAL	qqqREINTEGRO	tomar final grippe REINTEGRO	Certificado Aptitud	Visualizar Eliminar Imprimir Consulta

Items per page: 10 0 of 0

Figura 4.30: Lista Consultas Reintegro

Finalmente, el módulo tiene la opción de imprimir en formato PDF todos los datos correspondientes a una Consulta Reintegro que el médico haya seleccionado para su respectiva visualización.

4/1/22, 12:51 AM CONSULTA MEDICA PERIODICA

DATOS DEL PACIENTE

Empresa: FARMASOL EP
 Cédula: 0107012700
 Paciente: Christian Fernando Arizaga I
 Género: MASCULINO
 Cargo Paciente: CONTADOR

DESCRIPCIÓN

MOTIVO DE CONSULTA: PERIODICA
 ANTECEDENTES PERSONALES: FERNANDOODO IqPERIODICA

E. Hábitos tóxicos

	Tiempo consumo (meses)	Cantidad	Ex consumidor	Tiempo de Abstinencia (meses)
TABACO:	<input type="text"/>	<input type="text" value="0"/>	<input type="text"/>	<input type="text"/>
ALCOHOL:	<input type="text"/>	<input type="text" value="0"/>	<input type="text"/>	<input type="text"/>
OTRAS DROGAS	<input type="text"/>	<input type="text" value="0"/>	<input type="text"/>	<input type="text"/>

F. ESTILO DE VIDA

ACTIVIDAD FÍSICA: ¿CUÁL?: Descripción... TIEMPO / CANTIDAD:

MEDICACIÓN HABITUAL: ¿CUÁL?: Descripción... TIEMPO / CANTIDAD:

Figura 4.31: Consulta Reintegro

4.1.10. MÓDULO ACCIDENTES

Este módulo permitirá el registro de datos correspondientes a un Accidente que haya sufrido el paciente en su ámbito laboral, para ello se mostrará en un listado todos los pacientes registrados pudiendo el médico filtrar por el nombre correspondiente a ser atendido y así poder registrar un nuevo Accidente llenando los parámetros solicitados.

REGISTRAR ACCIDENTE

Paciente:
0107012700 Christian Fernando Arizaga Idrovo

Fecha Accidente: 01/04/2022 Hora Accidente: 12:58:03 AM Fecha Atención: 01/04/2022 Hora Atención: 12:58:03 AM

Detalle Medico | Detalle Seguridad Integral

Region Afectada: Parte Afectada: Naturaleza Lesion: Dias Incapacidad: Escala:

Supervisor: Reincidencia: Enviado A: Tipo de accidente: Grado de accidente:

Medico Atiende: Observaciones:

Figura 4.32: Registrar Accidente

Así podrá luego observar mediante un listado todos los accidentes sufridos un paciente en la empresa pudiendo si así lo requiere actualizar los datos del accidente seleccionado o también eliminar dicho accidente del registro.

LISTAR ACCIDENTE

Paciente:
0107012700 Christian Fernando Arizaga Idrovo

filter

Codigo	Fecha	Naturaleza Lesion	Nombre Region	PACIENTE	Parte Afecta	ACCIONES
1216	2022-03-14T05:00:00Z[UTC]	ESQUINCE	MIEMBRO DERECHO	CHRISTIAN FERNANDO	MIEMBRO IZQUIERDO	
1333	2022-03-21T05:00:00Z[UTC]	CABEZA	CABEZA	CHRISTIAN FERNANDO	CABEZA	

Items per page: 10 0 of 0

Figura 4.33: Lista Accidentes

Finalmente, el módulo tiene la opción de imprimir en formato PDF todos los datos correspondientes a un Accidente para su respectiva visualización.

4/1/22, 12:51 AM CONSULTA MEDICA PERIODICA

DATOS DEL PACIENTE

Empresa:	Cédula:	Paciente:	Género:	Cargo Paciente:
FARMASOL EP	0107012700	Christian Fernando Arizaga I	MASCULINO	CONTADOR

DESCRIPCIÓN

MOTIVO DE CONSULTA:	PERIODICA	ANTECEDENTES PERSONALES:	FERNANDOODO lo PERIODICA
---------------------	-----------	--------------------------	--------------------------

E. HÁBITOS TÓXICOS

	Tiempo consumo (meses)	Cantidad	Ex consumidor	Tiempo de Abstinencia (meses)
TABACO:	<input type="text"/>	<input type="text" value="0"/>	<input type="text"/>	<input type="text"/>
ALCOHOL:	<input type="text"/>	<input type="text" value="0"/>	<input type="text"/>	<input type="text"/>
OTRAS DROGAS	<input type="text"/>	<input type="text" value="0"/>	<input type="text"/>	<input type="text"/>

F. ESTILO DE VIDA

ACTIVIDAD FÍSICA:	¿CUÁL?:	Descripción..	TIEMPO / CANTIDAD:
MEDICACIÓN HABITUAL:	¿CUÁL?:	Descripción..	TIEMPO / CANTIDAD:

Figura 4.34: Accidente

4.1.11. HISTORIAL

Este módulo permite observar mediante un listado el historial correspondiente a un paciente que han sido registrados en cada una de las consultas realizadas anteriormente.

Para ello solo tendrá que buscar de lista al paciente del cual requiere y así acceder a todos los historiales, también existe la opción de poder visualizar todos los datos correspondientes solamente al último historial.

4.1.12. CERTIFICADO DE APTITUD

Por cada una de las consultas realizadas como lo son Preocupacional, Periódica, Retiro y Reintegro se deberá registrar un certificado de aptitud el cual estará relacionado dependiendo de la consulta realizada para lo cual esta interfaz permitirá el registro del mismo llenando todos los datos necesarios, posteriormente luego de haber realizado el certificado se lo podrá imprimir en formato PDF.

REGISTRAR CERTIFICADO DE APTITUD

[VOLVER ATRÁS](#) [IMPRIMIR CERTIFICADO DE APTITUD](#)

A. DATOS DEL ESTABLECIMIENTO - EMPRESA Y USUARIO

EMPRESA	CÉDULA:	PACIENTE:	FECHA DE INGRESO:	CARGO:
FARMASOL EP	0105729412	carolina moscoso	11/03/2022	CONTADOR

B. DATOS GENERALES

FECHA DE EMISIÓN: 14/03/2022

EVALUACION: INGRESO

C. CONCEPTO PARA APTITUD LABORAL

APTO:
 APTO EN OBSERVACIÓN:
 APTO CON LIMITACIONES:
 NO APTO:

OBSERVACIÓN: ES APTO PARA EL TRABAJO DESIGNADO

Figura 4.35: Registrar Certificado Aptitud

CERTIFICADO DE APTITUD

11/22, 1:03 AM

A. DATOS DEL ESTABLECIMIENTO - EMPRESA Y USUARIO

EMPRESA	CÉDULA:	PACIENTE:	FECHA DE INGRESO:	CARGO:
FARMASOL EP	0105729412	carolina moscoso	11/03/2022	CONTADOR

B. DATOS GENERALES

FECHA DE EMISIÓN: 14/03/2022

EVALUACION: INGRESO

C. CONCEPTO PARA APTITUD LABORAL

APTO:
 APTO EN OBSERVACIÓN:
 APTO CON LIMITACIONES:
 NO APTO:

OBSERVACIÓN: ES APTO PARA EL TRABAJO DESIGNADO

D. CONDICIONES DE SALUD AL MOMENTO DEL RETIRO

CONDICION DE SALUD: SATISFACTORIO

OBSERVACIONES: ok

Figura 4.36: Certifico Aptitud

4.1.13. MÓDULO REPORTES

Este módulo fue creado principalmente para que el médico pueda obtener un reporte con toda la información relevante de cada uno de los módulos que solicite implementados anteriormente que hayan sido generados o registrados en el sistema recibiendo como parámetros rangos de fechas (Fecha Inicio y Fecha Final) a excepción del módulo “**USUARIOS DEL SISTEMA Y PACIENTES**” de los cuales no es necesario generar un reporte.

Para ello solo se indicará el rango de fechas mencionadas y luego se presentará una lista de la información según el módulo seleccionado en formato PDF.

REPORTE DE ACCIDENTES POR FECHA

Fecha Inicio: Fecha Fin: Accidentes

Codigo	Fecha Accidente	Naturaleza Lesion	Region Afectada	Parte Afectada	TipoAccidente	Escala	Enviado A	Tipo Accidente	Paciente
--------	-----------------	-------------------	-----------------	----------------	---------------	--------	-----------	----------------	----------

GENERAR PDF

Figura 4.37: Reporte Módulo Accidentes

Para ello se consume un servicio REST desde el servidor JAVA devolviéndonos los datos requeridos a ser mostrados en el reporte. En la ilustración podemos observar el servicio para Accidentes la misma lógica se maneja para los demás módulos.

Get Accidentes por Fecha

Parámetros de entrada

- **Fecha Inicio:** Fecha inicial de Consulta.
- **Fecha Fin:** Fecha Final

```
@GET
@Path("/listAccidentesReporteFecha")
@Produces("application/json;charset=utf-8")
public List<Accidente>getAccidentesPorFecha(@QueryParam("fechaInicio")String fechaInicio,@QueryParam("fechaFin")String fechaFin) {
    return aon.getAccidentesPorFecha(fechaInicio, fechaFin);
}
}
```

Figura 4.38: Método reporte Accidentes por Fecha

Get Consultas Médica por Fecha

Parámetros de entrada

- **Fecha Inicio:** Fecha inicial de Consulta.
- **Fecha Fin:** Fecha Final

```
@GET
@Path("/ReporteConsultasMedicaPorFecha")
@Produces("application/json;charset=utf-8")
public List<ConsultaMedica>getConsultasMedicaPorFecha(@QueryParam("fechaInicio")String fechaInicio,@QueryParam("fechaFin")String fechaFin) {
    return con.getConsultasMedicaPorFecha(fechaInicio, fechaFin);
}
```

Figura 4.39: Método reporte Consultas Médica por Fecha

REPORTE DE CONSULTA MÉDICA GENERAL

Fecha Inicio: Fecha Fin: [CONSULTA MÉDICA GENERAL](#)

PESO	TALLA	MASA CORPORAL	ESTADO NUTRICIONAL	MOTIVO CONSULTA	PROBELMA ACTUAL	DIAGNOSTICO	TRATAMIENTO	PACIENTE
------	-------	---------------	--------------------	-----------------	-----------------	-------------	-------------	----------

[GENERAR PDF](#)

Figura 4.40: Reporte Módulo Consultas Médica

Get Consultas Preocupacional por Fecha

Parámetros de entrada

- **Fecha Inicio:** Fecha inicial de Consulta.
- **Fecha Fin:** Fecha Final

```
@GET
@Path("/ReporteConsultasPreocupacionalPorFecha")
@Produces("application/json;charset=utf-8")
public List<ConsultaPreocupacional> getConsultasPreocupacionalPorFecha(@QueryParam("fechaInicio")String fechaInicio,@QueryParam("fechaFin")String fechaFin) {
    return preocupacional0n.getConsultasPreocupacionalPorFecha(fechaInicio, fechaFin);
}
```

Figura 4.41: Método reporte Consultas Inicio-Preocupacional por Fecha

REPORTE DE CONSULTA PREOCCUPACIONAL / INICIO

Fecha Inicio: Fecha Fin: [CONSULTA PREOCCUPACIONAL](#)

MOTIVO CONSULTA	ACTIVIDAD EXTRALABORAL	ENFERMEDAD ACTUAL	DIAGNÓSTICO	TRATAMIENTO	PESO	TALLA	MASA CORPORAL	ESTADO NUTRICIONAL<tbody>	PACIENTE
-----------------	------------------------	-------------------	-------------	-------------	------	-------	---------------	---------------------------	----------

[GENERAR PDF](#)

Figura 4.42: Reporte Módulo Consultas Inicio

Get Consultas Preventiva por Fecha

Parámetros de entrada

- **Fecha Inicio:** Fecha inicial de Consulta.
- **Fecha Fin:** Fecha Final

```
@GET
@Path("/ReporteConsultasPreventivaPorFecha")
@Produces("application/json;charset=utf-8")
public List<ConsultaPreventiva>getConsultasPreventivaPorFecha(@QueryParam("fechaInicio")String fechaInicio,@QueryParam("fechaFin")String fechaFin) {
    return cpon.getConsultasPreventivaPorFecha(fechaInicio, fechaFin);
}
```

Figura 4.43: Método reporte Consultas Periódica por Fecha

REPORTE DE CONSULTA PERIÓDICA

Fecha Inicio: Fecha Fin:

MOTIVO CONSULTA ENFERMEDAD ACTUAL DIAGNÓSTICO TRATAMIENTO INCIDENTES APTITUD MÉDICA MASA CORPORAL ESTADO NUTRICIONAL PACIENTE

Figura 4.44: Reporte Módulo Consultas Periódica

Get Consultas Retiro por Fecha

Parámetros de entrada

- **Fecha Inicio:** Fecha inicial de Consulta.
- **Fecha Fin:** Fecha Final

```
@GET
@Path("/ReporteExámenesMedicoRetiroPorFecha")
@Produces("application/json;charset=utf-8")
public List<ExamenMedicoRetiro>ReporteExámenesMedicoRetiroPorFecha(@QueryParam("fechaInicio")String fechaInicio,@QueryParam("fechaFin")String fechaFin) {
    return emrOn.getExámenesMedicoRetiroPorFecha(fechaInicio, fechaFin);
}
```

Figura 4.45: Método reporte Consultas Retiro por Fecha

REPORTE DE CONSULTA RETIRO

Fecha Inicio: Fecha Fin:

INICIO LABORES FECHA SALIDA TIEMPO DIAGNÓSTICO EVALUACIÓN OBSERVACIÓN TRATAMIENTO MASA CORPORAL ESTADO NUTRICIONAL PACIENTE

Figura 4.46: Reporte Módulo Consultas Retiro

Get Consultas Reintegro por Fecha

Parámetros de entrada

- **Fecha Inicio:** Fecha inicial de Consulta.
- **Fecha Fin:** Fecha Final

```
@GET
@Path("/ReporteConsultasReintegroPorFecha")
@Produces("application/json;charset=utf-8")
public List<ConsultaReintegro>getConsultasReintegroPorFecha(@QueryParam("fechaInicio")String fechaInicio,@QueryParam("fechaFin")String fechaFin) {
    return cron.getConsultasReintegroPorFecha(fechaInicio, fechaFin);
}
```

Figura 4.47: Método reporte Consultas Reintegro por Fecha

REPORTE DE CONSULTA REINTEGRO

Fecha Inicio: Fecha Fin:

FECHA SALIDA FECHA REINGRESO CAUSA SALIDA MOTIVO CONSULTA ENFERMEDAD ACTUAL DIAGNÓSTICO TRATAMIENTO PACIENTE

Figura 4.48: Reporte Módulo Consultas Reintegro

4.1.14. UI (User Interface)

Se consideró además para la interfaz de usuario integrar una barra de accesibilidad con cuatro componentes principales los cuales permitan modificar o alterar el contenido Web brindando así poder cambiar el aspecto por defecto del sistema web según lo requiere un usuario siendo uno de los mas importantes lo que se refiere a color y contraste.

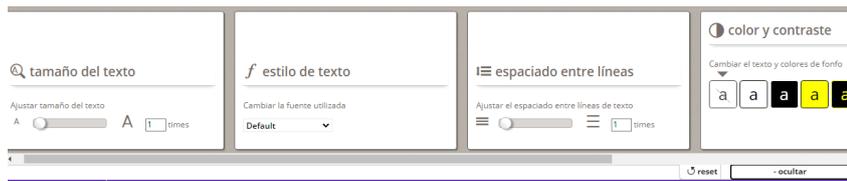


Figura 4.49: Barra de Accesibilidad

Para ello el primer componente permite ajustar el tamaño de texto de todas las etiquetas ayudando sobre todo a aumentar la letra para una mejor visión.



Figura 4.50: Ajustar Tamaño del Texto

El segundo componente permite cambiar el estilo de letra para ello existen 5 tipos de estilo que el usuario puede seleccionar y aplicar observando como todas las etiquetas o texto cambian su estilo.

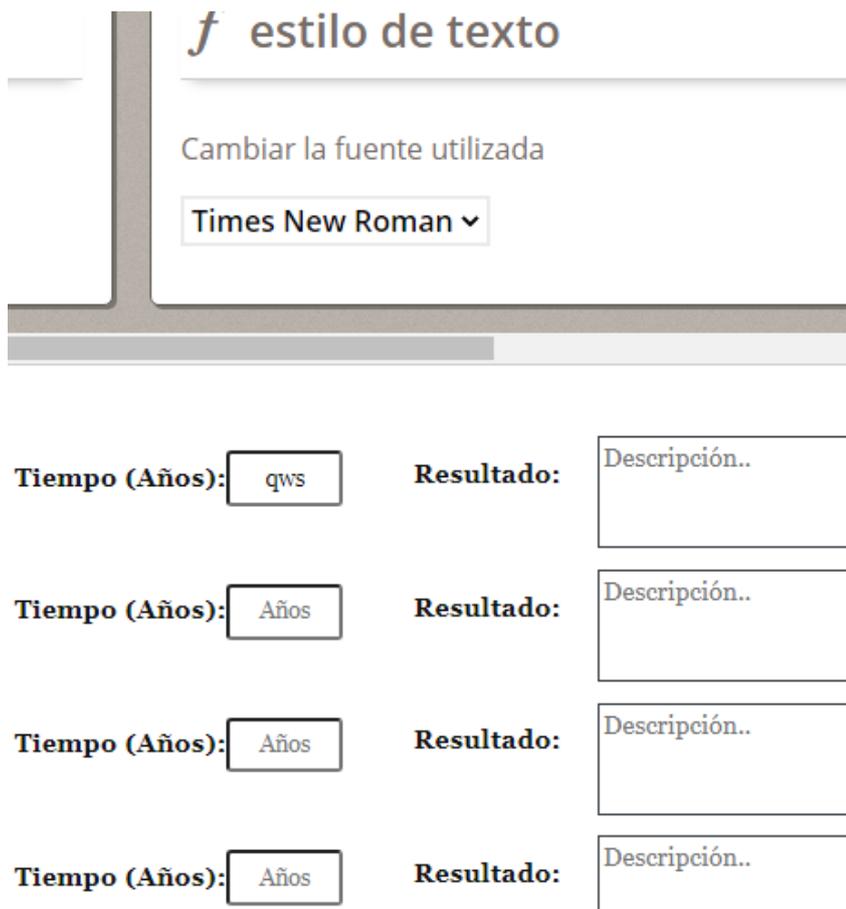


Figura 4.51: Componente Estilo de Fuente

El tercer componente permite cambiar el contraste de todos los elementos existentes en el sistema web es decir elegir el contraste entre texto y fondo de colores básicos separando el fondo del primer plano (color de la fuente) con los cuales mejora la accesibilidad de los mismos en la pantalla del ordenador ayudando en el enfoque visual del usuario final y percibir de una manera accesible los elementos del sistema web.

El cuarto componente esta enfocado principalmente a elementos como lo son botones y enlaces permitiendo realzar los mismos a través del aumento de tamaño de estos elementos si así lo requiera el usuario.

Sistema Médico Ocupacional

REGISTRAR CONSULTA PREOCUPACIONAL

Cédula: Paciente: Fecha Ingreso: Cargo Paciente:

PREOCUPACIONAL INICIO 1-3 PREOCUPACIONAL INICIO 2-3 PREOCUPACIONAL INICIO 3-3

A. DESCRIPCIÓN

MOTIVO DE CONSULTA: ANTECEDENTES PERSONALES:

B. ANTECEDENTES GINECO OBSTÉTRICOS

Menarquia: Ciclos: F.U.M: Gestas: Partos: Cesáreas:

Abortos: Vida Sexual Activa

Figura 4.52: Componente Ajustar Contraste

4.2. Análisis de Resultados

- En resumen se cumple con el objetivo general propuesto el de contar con un sistema de Información accesible de medicina ocupacional brindando a la empresa y al médico en si un nuevo sistema del manejo de la información de cada uno de sus trabajadores que luego podrá ser analizada y tratada para los respectivos tratamientos y prevención de enfermedades dentro del ambiente laboral.
- Se ha investigado sobre la accesibilidad en sistemas web como lo es en páginas oficiales permitiendo comprender como funcionan las páginas accesibles en internet y que tecnologías son necesarias para llegar a la mayor parte de usuarios manteniendo un correcto marcado e etiquetado de los elementos que brinden a herramientas de asistencia información relevante del contenido.
- Con lo que respecta al segundo objetivo el mantener una cercanía con el medico ocupacional perteneciente a la empresa ha sido de gran ayuda para cumplir con las funcionalidades debido a la correcta definición de los requisitos planteados permitiendo plasmarlos a través de diagramas tanto para la parte lógica como en la parte visual a través del diseño de la página manteniendo un sistema accesible que sea fácil de usar e interpretar cada una

de las funcionalidades dentro de un módulo.

- En lo que se refiere al desarrollo del sistema el mantener una estructura distribuida n-capas ha permitido un desarrollo limpio ya que cada una de las capas mantienen su funcionalidad como acceso a datos, lógica de negocio y presentación brindando un proyecto robusto y escalable separando la parte visual de todo el servidor back-end. Así al mantener el front-end separado permitió una mejor implementación de la tecnología WAI-ARIA(Web Accessibility Initiative Accessible Rich Internet Application) para mantener un sistema web accesible útil y dinámico para la interacción con el usuario final adecuándose a la actualidad en el desarrollo de páginas web que no presentes barreras o dificulten su acceso futuro a usuarios con algún tipo discapacidad específicamente visual.

- Por último se termina con la documentación a través de las pruebas del uso del sistema por parte del médico ocupacional de la empresa Farmasol como también con los manuales de instalación-despliegue y manual de usuario. Ver los anexos.
 - **Anexo A. Pruebas del sistema con el Médico Ocupacional y herramientas de Accesibilidad WAVE e IBM ACCESIBILITY CHECKER.**
 - **Anexo B. Manual de configuración y despliegue**
 - **Anexo C. Manual de Usuario**

Capítulo V

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- Concluyendo con el presente proyecto podemos indicar que la elaboración de este sistema dentro de la rama como es la medicina ocupacional se realizó previamente el levantamiento de los requerimientos funcionales conjuntamente con el medico ocupacional de FARMASOL, el mismo que nos permitió documentar las necesidades que presentaba el mismo, obteniendo así los módulos que serán necesarios para el desarrollo de software, el mismo que tiene como funcionalidad realizar una consulta médica y las diferentes consultas preocupacionales, de la misma forma el sistema nos permite realizar exámenes médicos ocupacionales lo mismos que son actos médicos que buscan el diagnostico de las condiciones de salud del trabajador, teniendo como objetivo: Definir la aptitud física y mental de los trabajadores para desempeñar un cargo que van a ser contratados. Obtener un diagnóstico claro de las condiciones de salud existentes en el trabajador mediante un examen de Inicio en el cual nuestro sistema cuenta con dicho módulo para realizar a un trabajador de la empresa llevando así con control de cada uno de ellos.
- Durante el estudio se empleó diferentes metodologías de investigación cuantitativas dentro del campo, el mismo que nos ayudó a conocer los aspectos importantes referente a la

accesibilidad profundizando en las personas con discapacidad visual, en Ecuador se tiene un porcentaje del 11.54 por ciento de la población que presenta este tipo de discapacidad, es por ello que en la actualidad los desarrolladores de páginas web sitúan énfasis en la creación de plataformas donde implementan tecnologías para la accesibilidad orientada justamente a personas con discapacidad, poniendo así en práctica en nuestro proyecto dicha característica en la cual se utilizó la tecnología WAI-ARIA permitiendo utilizar los atributos que brinda la misma, es por ello que nuestro sistema tendrá esa característica es decir, que un usuario podrá utilizar la plataforma mediante teclado entrando así en funcionamiento ARIA el mismo que presenta la integración a nuestro software, permitiendo así un lector de pantalla como puede ser NVDA leer todas las etiquetas integradas dentro de nuestro código, el mismo que contiene toda la información necesaria para el usuario que está utilizando la plataforma.

- Hay una gama de diferentes especificaciones relacionadas con WAI-ARIA cada uno de sus atributos tiene un propósito diferente como de igual forma su objetivo. Las especificaciones técnicas WAI-ARIA es principalmente para desarrolladores que trabajen en navegadores web, tecnologías de asistencia y herramientas de evaluación de accesibilidad. Las prácticas de creación de WAI-ARIA describen como los desarrolladores web pueden realizar aplicaciones accesibles es decir orientado a personas con discapacidad, mediante la utilización de esta gran herramienta como lo es ARIA.

Capítulo VI

REFERENCIAS BIBLIOGRAFICAS

6.1. Bibliografía

- Violeta María García Tijerino, M. (2014). ACCIDENTES LABORALES EN TRABAJADORES DE ZONA FRANCA. Nicaragua: Violeta.
- Burton, J. (2010). WHO healthy workplace framework and model: background and supporting literature and practices. <https://apps.who.int/iris/handle/10665/113144>
- Tuček, M. (2018). Occupational medicine and disease prevention. *Casopis Lekarů Ceskych*, 157(8), 392–395. <https://pubmed.ncbi.nlm.nih.gov/30754974/>
- W3C. (2021). World Wide Web Consortium (W3C). <https://www.w3.org/>
- W3C. (2022a). Requisitos de accesibilidad para personas con baja visión. Requisitos de Accesibilidad Para Personas Con Baja Visión. <https://www.w3.org/TR/low-vision-needs/>
- W3C. (2022b). Web Content Accessibility Guidelines (WCAG) 2.0. Pautas de Accesibilidad Al Contenido Web (WCAG) 2.0. <https://www.w3.org/TR/WCAG20/intro>
- WAI-ARIA. (2022). Descripción general de WAI-ARIA | Iniciativa de Accesibilidad Web (WAI) | W3C. Descripción General de WAI-ARIA. <https://www.w3.org/WAI/standards-guidelines/aria/>

- Angular. (2021). <https://material.angular.io/>. Obtenido de <https://material.angular.io/design/1.6.0/overview.html#material-theming>
- Gonçalves, M. J. (13 de Octubre de 2021). <https://www.hiberus.com/>. Obtenido de <https://www.hiberus.com/que-es-angular-y-para-que-sirve/>: :text= Angular20es20un20Framework20de,de20mejoras20para20este20framework
- Hat, R. (8 de Mayo de 2020). <https://www.redhat.com/>. Obtenido de <https://www.redhat.com/es/topics/api/what-is-a-rest-api>
- Kurtz, J., Wortman, B. (2014). Designing the Sample REST API. In ASP.NET Web API 2: Building a REST Service from Start to Finish (pp. 21–29). Apress. <https://doi.org/10.1007/978-1-4842-0109-1-3>
- García, A. (2019). Modelo de programación web y bases de datos (UF2405) (5th ed.). https://www.editorialelearning.com/catalogo/media/iverve/uploadpdf/1526035413_UF2405_emo.pdf
- ORACLE. (2022). Java Platform, Enterprise Edition (Java EE) | Oracle Technology Network | Oracle. Java EE at a Glance. <https://www.oracle.com/java/technologies/java-ee-glance.html>
- Elrom, E. (2016). AngularJS. In Pro MEAN Stack Development (pp. 101–129). Apress. https://doi.org/10.1007/978-1-4842-2044-3_5
- PostgreSQL. (2022). PostgreSQL: The world’s most advanced open source database. <https://www.postgresql.org/>
- NVDA. (2022). NVDA en español. Comunidad Hispanohablante Del Lector de Pantalla NVDA. <https://nvda.es/>
- FS. (2022). JAWS® – Freedom Scientific. Freedom Scientific. <https://www.freedomscientific.com/products/software/jaws/>
- Abou-Zahra, S. (2008). Web Accessibility Evaluation. 79–106. https://doi.org/10.1007/978-1-84800-050-6_7
- WAVE. (2022). WAVE Web Accessibility Evaluation Tool. <https://wave.webaim.org/>
- IBM. (2022). Tools – IBM Accessibility. Equal Access Accessibility Checker. <https://www.ibm.com/able/toolkit/tools/>

- WildFly. (2022). WildFly. WildFly . <https://www.wildfly.org/>
- Sacoto, P., Tapia, C. (2020). www.academia.edu. Obtenido de www.academia.edu: <https://www.academia.edu/422>
- genos. (2021). genos.es. Obtenido de genos.es: <https://genos.es/wildfly/>
- Arsys. (8 de Marzo de 2017). arsys.es. Obtenido de arsys.es: <https://www.arsys.es/blog/programacion/wildfly-cloud>.
- Hat, R. (8 de Mayo de 2020). redhat.com. Obtenido de redhat.com: <https://www.redhat.com/es/topics/api/what-is-a-rest-api>.

Anexos

ANEXO A

RESULTADOS

HERRAMIENTA DE EVALUACIÓN DE ACCESIBILIDAD WEB WAVE

Es una herramienta en el cual nos ayuda a los desarrolladores de paginas web medir su nivel de accesibilidad para personas con discapacidad, permitiendo asi poder identificar posibles fallos o errores que se tenga en dicha pagina, haciendo uso de esta herramienta realizamos nuestras pruebas de nuestro sistema y se obtuvo los siguientes resultados.

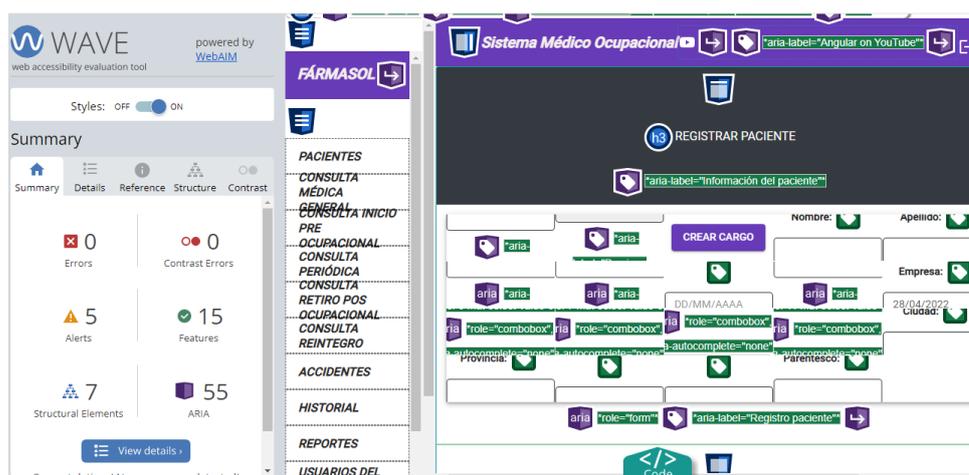


Figura 0.1: Pruebas con la herramienta WAVE



Figura 0.2: Pruebas con la herramienta WAVE

De igual forma se utilizó de una herramienta como es IBM Equal Access Accessibility Checker en la cual también permite medir lo que en nivel de accesibilidad de nuestra plataforma web, esta herramienta permite identificar los errores y como también sugerencias para darle solución a los mismos. Los resultados son los siguientes.

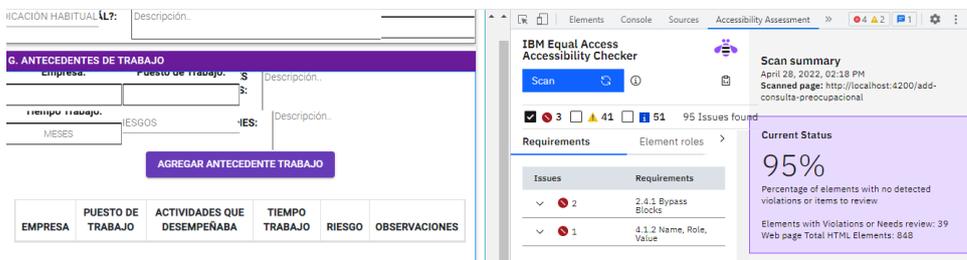


Figura 0.3: Pruebas con la herramienta IBM Equal

También se realizó pruebas con una persona no vidente para su respectiva evaluación de la página web, en las cuales los resultados fueron exitosos ya que pudo manejar sin dificultad alguna nuestro sistema como se puede apreciar en las siguientes imágenes.

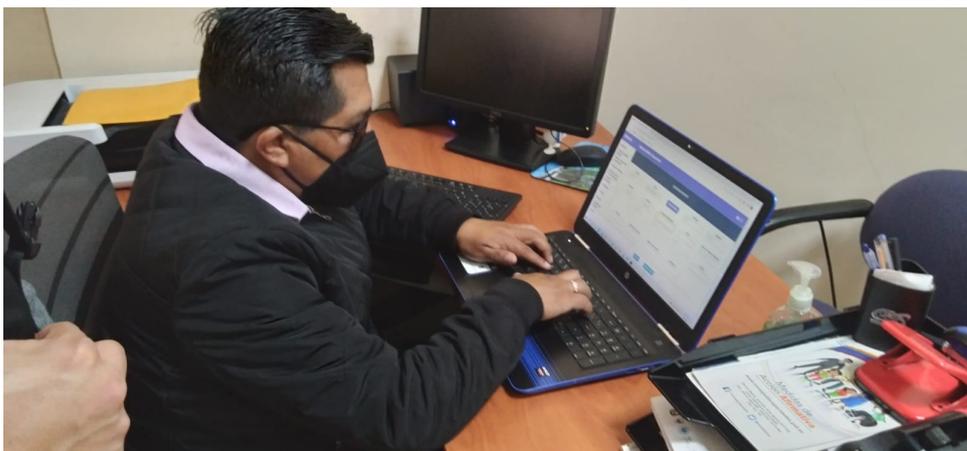


Figura 0.4: Pruebas persona no vidente



Figura 0.5: Pruebas persona no vidente

De igual forma se realizó pruebas de funcionamiento de nuestro sistema con la Doctora Carolina Moscoso encargada del departamento de la medicina ocupacional de la empresa de Farmasol, en la cual el resultado fue correcto ya que cumple con cada uno de los requisitos que lo solicitaban .

ANEXO B

Manual de Instalación y Despliegue

Manual de Instalación y Despliegue correspondiente al sistema Medico Ocupacional para ello se observa los requerimientos mínimos para que el sistema se pueda ejecutar de una manera correcta.

Todo el proceso de Despliegue se lo realizara en un servidor Linux concretamente haciendo uso del Sistema Operativo Ubuntu 20.

Carácterística	Linux
Sistema	Reportes
Espacio en disco	Doctor
Herramientas	- Ingreso de fecha desde hasta para visualizar e exportar en PDF. - Visulaizar Reportes.
Memoria	Validar fechas ingresadas. presentación de los datos obtenido en un PDF
Procesador	ALTA
Navegadores:	Goolge Chroome o Mozilla Firefox

Tabla 0.1: Requerimientos del Sistema

A continuación, se detalla el proceso de despliegue para que el sistema funcione correctamente en un entorno Linux específicamente en el sistema operativo Ubuntu Server 18.

JAVA JDK

Lo primero y fundamental es tener nuestro sistema actualizado para evitar errores o problemas de dependencias en el proceso de despliegue para ello ejecutaremos los siguientes comandos en modo administrador:

```
"sudo apt-get update; apt-get upgrade"
```

Posteriormente instalaremos una herramienta para descomprimir todos nuestros archivos .zip que vayamos necesitando en el despliegue para ello ejecutaremos lo siguiente:

```
"sudo apt-get install zip unzip"
```

Luego deberemos tener instalado el paquete OpenJDK para ello lo descargaremos en un directorio especificado y así poder tener en el sistema todas sus dependencias todo esto lo haremos ejecutando el siguiente comando en la terminal:

```
"sudo apt-get install default-jdk"
```

Verificamos que se haya instalado el paquete correctamente usando el siguiente comando:

```
java -version
```

```
      # java -version
java version "1.7.0_201"
OpenJDK Runtime Environment (IcedTea 2.6.17) (7u211-2.6.17-0ubuntu0.1)
OpenJDK 64-Bit Server VM (build 24.201-b00, mixed mode)
```

Figura 0.6: Comprobación paquete instalado

BASE DE DATOS: POSTGRESQL

Descargaremos los paquetes para contar con nuestra base de datos a usar en este caso postgresql necesarios para poder persistir nuestra información y lo ubicamos en una carpeta en específico para ello usamos el siguiente los siguientes comandos:

```
sudo apt-get install postgresql postgresql-contrib -y
```

```
sudo apt-get install postgresql-client -y
```

Lo siguiente será proceder a ingresar a nuestra base ejecutando el siguiente comando “sudo su postgres” y escribir lo siguiente “psql” para ubicarnos dentro de la base, así crearemos nuestro usuario administrador del sistema gestor de base de datos el cual será postgres ejecutando el siguiente comando:

```
alter user postgres with password 'Admin@123';
```

Luego crearemos nuestra respective base en cual persistirá los datos siendo el propietario nuestro usuario postgres.

```
create database postgres owner postgres;
```

AL usuario creado le asignamos el rol de superusuario y a la vez le asignamos todos los privilegios necesarios sobre la base de datos a usar empleando los siguientes comandos:

```
alter role postgres with superuser;
```

```
grant all privileges on database postgres to vis;
```

Verificaremos todo lo anterior y podremos observar como la base de datos se ha creado y que el propietario es nuestro usuario postgres.

```
es
List of databases
Name | Owner | Encoding | Collate | Ctype | Access privileges
-----+-----+-----+-----+-----+-----
postgres | postgres | UTF8 | es_EC.UTF-8 | es_EC.UTF-8 | =Tc/postgres
postgres template0 | postgres | UTF8 | es_EC.UTF-8 | es_EC.UTF-8 | =c/postgres
postgres template1 | postgres | UTF8 | es_EC.UTF-8 | es_EC.UTF-8 | =c/postgres
(3 rows)
```

Figura 0.7: Base de Datos Creada

WILDFLY

WildFly como tal es un servidor de aplicaciones compatible con JAVA JEE, también es Open Source es decir no tiene ningún costo alguno, teniendo así grandes ventajas como su flexibilidad, ligero y administrativo en la creación e implementación de aplicaciones y servicios JAVA.

Para ello será necesario instalarlo en el servidor ubuntu y poder así implementar nuestro servidor backend que proveerá de todos los servicios para el sistema web.

Así procedemos a instalar el servidor de aplicaciones deberemos de igual forma descargar el paquete Wildfly para ello lo haremos descargando el archivo .zip de las fuentes oficiales y ubicándolo respectivamente en la carpeta “/opt” que es la carpeta donde estará alojado, luego se lo descomprime moviéndolo después a la carpeta Wildfly y finalmente eliminamos el “.zip” que ya no es necesario, todo esto lo haremos ejecutando los siguientes comandos:

```
"sudo groupadd -r wildfly"
```

```
"sudo useradd -r -g wildfly -d /opt/wildfly -s /sbin/nologin wildfly"
```

```
cd /opt"
```

```
"sudo wget https://download.jboss.org/wildfly/24.0.1.Final/wildfly-24.0.1.Final.zip"
```

```
"sudo unzip wildfly-24.0.1.Final.zip"
```

```
"sudo mv wildfly-24.0.1.Final wildfly"
```

```
"sudo rm wildfly-24.0.1.Final.zip"
```

Luego procederemos a generar variables del entorno wildfly requeridos para una conexión hacia el servidor para ello editaremos el archivo “wildfly” agregando las siguientes variables:

```
"sudo nano /etc/default/wildfly"
```

```
WILDFLY_USER=" wildfly "  
"STARTUP_WAIT=180"  
"SHUTDOWN_WAIT=30"  
"WILDFLY_CONFIG=standalone.xml"  
"WILDFLY_MODE=standalone "  
"WILDFLY_BIND=0.0.0.0 "
```

A continuación, se procede a editar el archivo bash launch.sh lo cual permitirá que al encender el servidor ubuntu arranque automáticamente el servidor wildfly sin necesidad de hacerlo manualmente, así agregamos las siguientes líneas y variables:

```
"sudo nano /opt/wildfly/bin/launch.sh"
```

```
" #!/bin/sh  
if [ "x$WILDFLY_HOME" = "x" ];then  
    WILDFLY_HOME=/opt/wildfly  
fi  
if [ "x1" = "xdomain" ]; then  
    echo 'Starting Wildfly in domain mode.'
```

```

        $WILDFLY_HOME/bin/domain.sh -c $2 -b $3
else
        echo 'Starting Wildfly in standalone mode.'
        $WILDFLY_HOME/bin/standalone.sh -c $2 -b 3
fi
"

```

Asignamos los permisos necesarios para el archivo launch.sh.

```
"sudo chmod 755 /opt/wildfly/bin/launch.sh"
```

Editamos el archivo wildfly.service agregando las siguiente variables de entorno:

```
"sudo nano /etc/systemd/system/wildfly.service"
```

```

[ Unit ]
Description=The WildFly Application Server
After=syslog.target network.target
Before=nginx.service

[ Service ]
Environment=LAUNCH_JBOSS_IN_BACKGROUND=1
EnvironmentFile=/etc/default/wildfly
User=wildfly
LimitNOFILE=102642
PIDFile=/var/run/wildfly/wildfly.pid

```

```
ExecStart=/opt/wildfly/bin/launch.sh $WILDFLY_MODE  
$WILDFLY_CONFIG WILDFLY_BIND  
StandardOutput=null
```

```
[ Install ]
```

```
WantedBy=multi-user.target
```

Asignamos permisos al usuario wildfly sobre la carpeta wildfly para poder realizar todas las acciones sobre ella y finalmente reiniciamos el servicio y lo habilitamos para que este disponible y listo para su uso, esto lo haremos ejecutando los siguientes comandos:

```
"sudo chown wildfly:wildfly -R /opt/wildfly/"
```

```
"sudo systemctl daemon-reload"
```

```
"sudo systemctl start wildfly"
```

```
"sudo systemctl enable wildfly"
```

Finalmente, usando el siguiente comando crearemos el usuario administrador de Wildfly con su respectiva contraseña, esto es útil para poder desplegar nuestro proyecto en el servidor ya que lo haremos desde la consola de administración de wildfly.

```
"sudo bash /opt/wildfly/bin/add-user.sh"
```

```
"What type of user do you wish to add?"
```

```
.a) Management User (mgmt-users.properties)"
```

```
"b) Application User (application-users.properties)"
```

"(a):a "

A continuación, copiaremos la carpeta **ORG** la cuál cuenta con el driver y todos los módulos necesarios de postgresql.

Copiar las librerías JDBC como módulos del jboss. Copiar el contenido del comprimido **modules.zip** que usará el servidor para la conexión respectiva con

la base de datos para ello copiamos dicha carpeta en la carpeta modules de nuestro JBOSS_HOME ejecutaremos el siguiente comando:

```
"sudo unzip modules"
```

```
"sudo cp -R /org /opt/wildfly/modules"
```

Se edita el archivo standalone.xml de wildfly para agregar el datasource acorde a la base de datos útil para la conexión desde el servidor hacia la base, a su vez agregaremos las variables respectivas como:

Nombre base de Datos: postgres

user-name: postgres

password: Admin@123

```

<datasources>
  <datasource jndi-name="java:jboss/datasources/ExampleDS" pool-name="ExampleDS" enabled="true" use-java-context="true" statistics-enabled="${wildfly.d
  <connection-url>jdbc:h2:mem:test;DB_CLOSE_DELAY=-1;DB_CLOSE_ON_EXIT=FALSE</connection-url>
  <driver>h2</driver>
  <security>
    <user-name>sa</user-name>
    <password>sa</password>
  </security>
</datasource>
  <datasource jta="true" jndi-name="java:jboss/datasources/ejPostgresDS" pool-name="ejPostgresDS" enabled="true" use-java-context="true" use-ccm="true">
  <connection-url>jdbc:postgresql://localhost:5432/postgres</connection-url>
  <driver>postgresql-driver</driver>
  <security>
    <user-name>postgres</user-name>
    <password>patito.123</password>
  </security>
</datasource>
</drivers>
</datasources>

```

Figura 0.8: Datasource

Agregar en la seccion <drivers>....</drivers> del standalone.xml el driver de postgresql especificado.

```

"<driver name=" postgresql - driver " module=" org . postgresql ">
  <driver - class >org . postgresql . Driver </ driver - class >
</ driver >"

```

Verificamos que wildfly se encuentre activo para ello realizamos un reinicio del servicio y observamos su estado ejecutando el siguiente comando:

"sudo systemctl restart wildfly"

"sudo systemctl status wildfly"

```

.# systemctl restart wildfly
# systemctl status wildfly
● wildfly.service - The WildFly Application Server
   Loaded: loaded (/etc/systemd/system/wildfly.service; enabled; vendor preset:
   Active: active (running) since mié 2022-03-30 17:53:36 -05; 10s ago
   Main PID: 4087 (launch.sh)
   CGroup: /system.slice/wildfly.service
           └─4087 /bin/sh /opt/wildfly/bin/launch.sh standalone standalone.xml 0
             └─4088 /bin/sh /opt/wildfly/bin/standalone.sh -c standalone.xml -b 0.
               └─4148 java -D[Standalone] -server -Xms64m -Xmx512m -XX:MetaspaceSize

mar 30 17:53:36 srv-ubuntu systemd[1]: Started The WildFly Application Server.
lines 1-10/10 (END)

```

Figura 0.9: Comprobación estado Wildfly

Finalmente desplegamos nuestro proyecto con la extensión .war desde la consola de administración usando nuestras credenciales creadas anteriormente subiendo nuestro proyecto en el apartado Deployments.

Así nuestro front-end podrá hacer uso de los distintos servicios creados en el servidor back-end.

ANEXO C

MANUAL DE USUARIO

- Ventana de Login

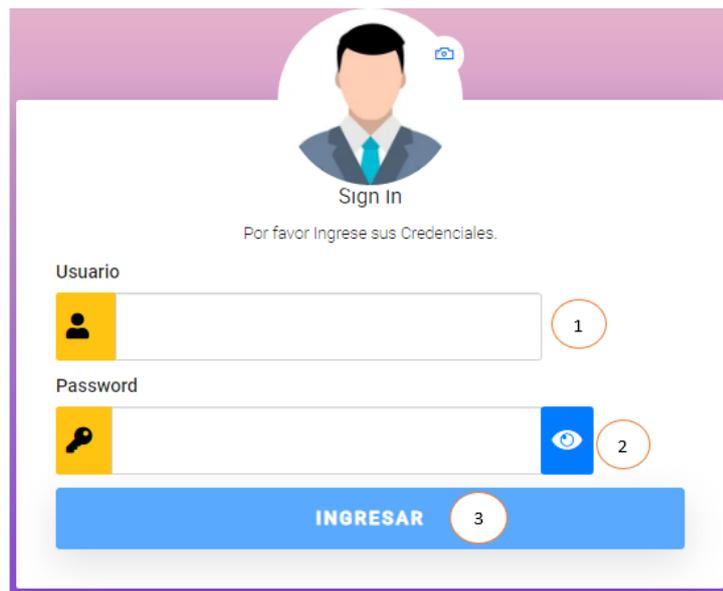


Figura 0.10: Login

1. "Ingreso de usuario."
2. "Ingreso de contraseña,"
3. Botón de Ingreso el mismo que valida si sus credenciales son correctas permite el ingreso de las diferentes opciones que tiene el sistema, caso contrario muestra un mensaje de Credenciales Incorrectas.

- Ventana principal



Figura 0.11: Ventana principal

1. Toolbar horizontal donde presenta información como nombre de la empresa y un título de un sistema. .
2. Botones de opciones, según el médico ocupacional lo solicite.

- Ventana Pacientes

1. Botón de ingreso para pacientes el mismo que me permite abrir una venta donde se puede visualizar una lista de pacientes ingresados anteriormente.
2. Lista de pacientes Ingresados anteriormente.
3. Botón que permite crear nuevo paciente.

CODIGO	CEZULA	NOMBRE	APELLIDO	EDAD	SEXO	TIPO SANGRE	TELEFONO	ACCIONES
1	0107012700	Christian Fernando	Arizaga Idrovo	27	MASCULINO	A+	0984432028	Editar Eliminar
133	0105729412	carolina	moscoso	30	FEMENINO	A+	0998659945	Editar Eliminar
35	0105526406	Johnny Michael	Espinoza Heredia	25	MASCULINO	A-	0984432028	Editar Eliminar
1342	0107012700	William Ronaldo	Espinoza Heredia	27	MASCULINO	A+	09846245	Editar Eliminar

Figura 0.12: Lista pacientes

4. Columna de acciones donde me permite editar un paciente o eliminar un paciente.

- Agregar Pacientes

Figura 0.13: Agregar paciente

1. Formulario de registro de paciente.
2. Crear cargo en caso que no existiera.
3. Botón que permite guardar un paciente.
4. Permite regresar atrás sin guardar ninguna información.

- Ventana lista de pacientes para consulta médica general

LISTA DE PACIENTES PARA CONSULTA MÉDICA GENERAL									
Buscar									
	CODIGO	CEDULA	NOMBRE	APELLIDO	EDAD	SEXO	TIPO SANGRE	TELEFONO	ACCIONES
	1	0107012700	Christian Fernando	Arizaga Idrovo	27	MASCULINO	A+	0984432028	Agregar Consultas
	133	0105729412	carolina	moscoso	30	FEMENINO	A+	0998659945	Agregar Consultas
	35	0105526406	Johnny Michael	Espinoza Heredia	25	MASCULINO	A-	0984432028	Agregar Consultas
	1342	0107012700	William Ronaldo	Espinoza Heredia	27	MASCULINO	A+	09846245	Agregar Consultas

Figura 0.14: Lista de pacientes para consulta

1. Botón de consulta médica General me permite abrir una venta de una lista de pacientes donde el médico tiene la opción de agregar una nueva consulta médica o ver las consultas que a tenido dicho paciente.
2. Lista de pacientes.
3. Columna de acciones en los cuales el médico tiene dos opciones por paciente en la cual la primera es agregar una nueva consulta o la segunda es

que le permite visualizar las consultas las consulta que a realizado anteriormente un determinado paciente.

- Ventana de lista de consultas por paciente

CODIGO	FECHA	PESO	TALLA	MASA CORPORAL	ESTADO NUTRICIONAL	TEMPERATURA	MOTIVO CONSULTA	PROBLEMA ACTUAL	ACCIONES
1239	2022-03-15	70	1.65	25.71	SOBREPESO	78			Editar Eliminar Certificado Reposo
1244	2022-03-15	80	1.86	23.12	PESO NORMAL	43			Editar Eliminar Certificado Reposo

Figura 0.15: Lista de consultas por paciente

1. Permite visualizar la información del paciente.
2. Columna de acciones donde tenemos 3 opciones uno nos permite editar la consulta que se a realizado anteriormente, dos eliminar la consulta, tres Certificado de reposo en la cual nos genera un pdf con un certificado dependiendo a los días que el médico indique.
3. Ingresar una nueva consulta médica.
4. Volver atrás, permite volver a la lista de pacientes.

- Ventana de reportes

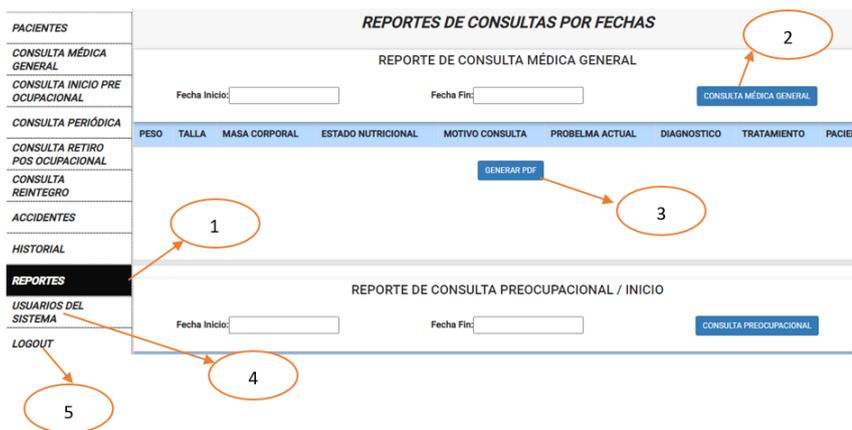


Figura 0.16: Reportes

1. Botón reportes: me permite obtener un reporte entre un rango de fechas de inicio y de fin de las consultas realizadas anteriormente.
2. Una vez ingresado las fechas desde hasta presionamos el botón consulta médica general y se nos genera el reporte en una tabla.
3. Una vez obtenido la información en la tabla con el botón GENEAR PDF nos genera un pdf con los datos previamente obtenido en la cual podemos imprimir.
4. Botón usuarios del sistema me permite crear un usuario el mismo que podrá autenticarse en la venta login para posteriormente tener acceso a las funciones del sistema.
5. Botón LOGUT me permite salir del sistema.



Figura 0.17: Barra de accesibilidad

- Barra de accesibilidad

1. El usuario tendrá una lista de opciones como por ejemplo aumentar el tamaño del texto.
2. Podrá darle un estilo al texto.
3. Un espaciado entre líneas.
4. Un color y contraste hacia toda la página.