



Construcción de un aplicativo web que permite ejecutar un plan de mantenimiento preventivo y correctivo de software y hardware de los equipos TIC'S en la radio stereo Latacunga ubicado en el cantón Latacunga.

Chicaiza Llumiquinga, Yadira Nataly y Guevara Miranda, Jonathan Xavier

Departamento de Eléctrica y Electrónica

Carrera de Tecnología Superior en Redes y Telecomunicaciones

Trabajo de Integración Curricular, previo a la obtención del título de Tecnólogo Superior en
Redes y Telecomunicaciones

Ing. Caiza Caizabuan, José Rubén, Mgrt.

08 de febrero de 2022

Latacunga

Reporte de Verificación de contenido

09/02, 15:08

JONATHAN XAVIER GUEVARA MIRANDA - Documento sin título

Informe de originalidad

NOMBRE DEL CURSO
Revisión Monografía

NOMBRE DEL ALUMNO
JONATHAN XAVIER GUEVARA MIRANDA

NOMBRE DEL ARCHIVO
JONATHAN XAVIER GUEVARA MIRANDA - Documento sin título

SE HA CREADO EL INFORME
30 ene 2023

Resumen

Fragmentos marcados	28	5 %
Fragmentos citados o entrecorridos	1	0,2 %

Coincidencias de la Web

lebeschool.com	4	0,7 %
amazon.com	4	0,7 %
plati.com	4	0,6 %
itpedia.nl	3	0,6 %
oracle.com	3	0,6 %
wearefresh.co	2	0,6 %
pmoinformatica.com	2	0,5 %
academia.edu	1	0,4 %
medium.com	2	0,2 %
tokioschool.com	1	0,1 %
cognos.com.bo	1	0,1 %
informatica.edu.bo	1	0,1 %
itdo.com	1	0,1 %



Ing. Caiza Caizabuano, José Rubén, Mgtr.

C.C.: 0502654296



Departamento de Eléctrica y Electrónica

Carrera de Tecnología Superior En Redes y Telecomunicaciones

Certificación

Certifico que el trabajo de Unidad de Integración Curricular: "Construcción de un aplicativo web que permite ejecutar un plan de mantenimiento preventivo y correctivo de software y hardware de los equipos TIC's en la radio stereo Latacunga ubicado en el cantón Latacunga" fue realizada por el/los señor/señores Chicaiza Llumiquinga Yadira Nataly, Guevara Miranda Jonathan Xavier, la misma que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisada y analizada en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se la sustente públicamente.

Latacunga, 08 de febrero del 2023

Ing. Caiza Caizabano José Rubén, Mgtr

C.C.: 0502854296



Departamento de Eléctrica y Electrónica

Carrera de Tecnología Superior En Redes y Telecomunicaciones

Responsabilidad de Autoría

Nosotros, Chicaiza Llumiquinga Yadira Nataly, con cédula de ciudadanía N° 05502706877, Guevara Miranda Jonathan Xavier con cédula de ciudadanía N° 1756282826, declaramos que el contenido, ideas y criterios del trabajo de Unidad de Integración Curricular "Construcción de un aplicativo web que permite ejecutar un plan de mantenimiento preventivo y correctivo de software y hardware de los equipos TIC's en la radio stereo Latacunga ubicado en el cantón Latacunga" es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Latacunga, 08 de febrero de 2022

Chicaiza Llumiquinga, Yadira Nataly

C.C.: 0550276877

Guevara Miranda, Jonathan Xavier

C.C.: 1756282826



Departamento de Eléctrica y Electrónica
Carrera de Tecnología En Redes Y Telecomunicaciones

Autorización de Publicación

Nosotros, Chicaiza Llumiquinga Yadira Nataly con cedula de ciudadanía N° 0550276877 y Guevara Miranda Jonathan Xavier con cedula de ciudadanía N° 1756282826, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de Unidad de Integración Curricular "Construcción de un aplicativo web que permite ejecutar un plan de mantenimiento preventivo y correctivo de software y hardware de los equipos TIC's en la radio stereo Latacunga ubicado en el cantón Latacunga" en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi/nuestra responsabilidad.

Latacunga, 08 de febrero del 2023

Chicaiza Llumiquinga, Yadira Nataly

C.C.: 0550276877

Guevara Miranda, Jonathan Xavier

C.C.: 1756282826

Agradecimiento

Agradezco en primer lugar a Dios por darme la fortaleza y sabiduría para alcanzar mis metas.

A mis padres, quienes siempre me han apoyado incondicionalmente en el logro de todas mis metas personales y académicas. Ellos son los que con su cariño me han impulsado siempre a perseguir mis metas y nunca abandonarlas frente a las adversidades. También son los que me han brindado el soporte material y económico para poder concentrarme en los estudios y nunca abandonarlos, además agradezco a mi noble institución por darme la oportunidad de seguir desarrollándome como profesional.

Hay tantos maestros que han sido parte de mi viaje universitario y quiero agradecerles a todos ellos por impartir el conocimiento necesario para estar aquí hoy, gracias a mi tutor por su dedicación y paciencia, sin sus palabras y correcciones precisas no hubiese podido lograr llegar a esta instancia tan anhelada.

Gracias a todos mis compañeros, muchos de los cuales se han convertido en mis amigos, compañeros y hermanos. Gracias por las horas, el trabajo en equipo y las historias vividas.

Chicaiza Llumiyinga, Yadira Nataly

Dedicatoria

El presente trabajo de titulación está dedicado a:

A mis padres Ramiro y Lilia quienes con su amor, paciencia y esfuerzo me han permitido llegar a cumplir hoy uno de mis sueños, gracias por inculcarme el ejemplo de esfuerzo y valentía, sin miedo ante las adversidades que se presenten porque Dios siempre está conmigo.

Mi hermano y mis hermanas por su amor y apoyo incondicional, durante todo este proceso, por estar conmigo en todo momento gracias. A toda mi familia porque con sus oraciones, consejos y ánimos me han hecho una mejor persona y de una forma u otra están conmigo apoyándome en cada uno de mis sueños y metas.

Finalmente quiero dedicar esta tesis a todas mis amigos y amigas, por apoyarme cuando más lo necesito, por ayudarme en los momentos difíciles y por todo el amor que me brindaron todos los días verdaderamente les agradezco de todo corazón. Muchas gracias siempre los llevare en mi corazón.

Chicaiza Llumiyinga, Yadira Nataly

Agradecimiento

En primer lugar, quisiera agradecer al tutor de esta monografía, Ing. Caiza José, por su dedicación y apoyo a este trabajo, por seguir mis sugerencias e ideas, y por su dirección y rigor. disponible para ellos.

Gracias por la confianza que me habéis mostrado a lo largo de mi carrera. También me gustaría agradecer a mis compañeros de la industria eléctrica y electrónica por su apoyo personal y humano a lo largo de los años, con quienes he compartido proyectos e ilusiones.

Gracias a mi familia, papá, porque con ellos pasé una infancia feliz, lo guardo en la memoria y también es un estímulo para mí para seguir escribiendo sobre mi infancia. Gracias a mis amigos, que siempre me han brindado un enorme apoyo moral y humano, que ha sido crucial en este momento tan difícil para este trabajo y esta profesión.

Guevara Miranda, Jonathan Xavier

Dedicatoria

El presente trabajo de titulación está dedicado a:

A Dios que me ha guiado hasta el día de hoy, su fuerza y su mano de lealtad y amor están siempre conmigo.

Gracias a mi padre, Freddy Manuel Guevara Paz, por su amor, paciencia y trabajo duro por hacerme alcanzar hoy un sueño más, gracias por inculcarme el trabajo y el coraje. Un ejemplo de no tener miedo a la adversidad porque Dios y mi padre siempre está conmigo.

Gracias a mi tía Angelita por brindarme su amor y apoyo incondicional durante todo este proceso, por acompañarme, gracias.

A toda mi familia porque sus oraciones, consejos y palabras de aliento me han hecho mejor persona y me han acompañado de alguna manera a alcanzar todos mis sueños y metas.

Finalmente, quisiera dedicar este increíble momento a una persona increíble que me acompañó en todo momento y que hoy día por circunstancias no está a mi lado Abigail estuvo detrás de mí cuando más la necesite. Sin duda eres la mejor persona que se ha podido cruzar en mi camino, eres mi apoyo.

Guevara Miranda, Jonathan Xavier

ÍNDICE DE CONTENIDOS

Carátula.....	1
Reporte de verificación de contenido.....	2
Certificación	3
Responsabilidad de autoría.....	4
Autorización de publicación.....	5
Agradecimiento	6
Dedicatoria.....	7
Agradecimiento	8
Dedicatoria.....	9
Índice de contenidos.....	10
Índice de tablas	17
Indice de figuras.....	18
Resumen	20
Abstract	21
Capítulo I: Introducción	22
Antecedentes	22
Planteamiento del Problema	23
Justificación	24
Objetivos	25
<i>Objetivo General</i>	25
<i>Objetivos Específicos</i>	25
Alcance.....	25
Capítulo II: Marco Teórico	27

Software	27
Licencias de Software	27
Categorías De Software	28
<i>Software Libre</i>	28
<i>Software Open Source</i>	28
Dominios de Aplicación del Software	29
<i>Software de Sistemas</i>	29
<i>Software de Aplicación</i>	29
<i>Software de Ingeniería y Ciencias</i>	29
<i>Software Incrustado</i>	30
<i>Software de Línea de Productos</i>	30
<i>Software de Inteligencia Artificial</i>	30
<i>Aplicaciones Web</i>	30
Tipos de Mantenimientos de un equipo informático	31
<i>Mantenimiento</i>	31
Mantenimiento preventivo.....	31
Mantenimiento correctivo	31
Mantenimiento correctivo de hardware	32
Mantenimiento correctivo de software.....	32
Lenguajes de programación	32
<i>Lenguaje de Merca de Hipertexto (HTML)</i>	33
<i>Hojas de Estilo en Cascada (CSS)</i>	33
<i>JavaScript</i>	33
<i>NodeJS</i>	34
<i>ReactJS</i>	34

<i>JSX</i>	34
Servidor web y servicios en la nube	34
<i>Servicio Web</i>	34
<i>Modelo Cliente Servidor</i>	35
<i>Servidor</i>	35
<i>Cliente</i>	35
<i>Servidor Node.js</i>	35
<i>Amazon Web Services</i>	36
Metodología de Desarrollo de Aplicaciones	36
<i>Metodología Scrum</i>	36
Antecedentes	36
Definición	36
Desarrollo de Software Agile	37
<i>Perfiles de la Metodología Scrum</i>	37
Product Owner.....	37
Scrum Master.....	37
Scrum Team.....	37
Stakeholder.....	37
<i>Etapas de la Metodología Scrum</i>	37
Etapa 1: Product Backlog	38
Etapa 2: Sprint Backlog.....	38
Etapa 3: Sprint Review	39
Etapa 4: Sprint Planning Meeting	39
Ciclo de vida de un proyecto de Software	40
Recopilación de los Requerimientos	40

Motor de búsqueda	42
Bases de datos	42
Modelos de SGBD	46
Arquitectura del software	47
Patrón de capas	48
Patrón master-slave.	49
Patrón modelo-vista-controlador (MVC)	49
Patrón broker	51
FrontEnd	51
BackEnd	52
Diferencias entre FrontEnd y Backend	52
Capítulo III: Desarrollo del tema	53
Metodología SCRUM	53
Análisis de Encuesta	56
<i>Encuesta Administrador y Personal de la Radio Latacunga FM-AM</i>	56
Requerimiento	60
<i>Requerimiento de negocio</i>	60
<i>Análisis del mercado</i>	61
<i>Funcionalidades</i>	61
Rol De Usuario	61
Rol De Administrador	62
Entorno Operativo	64
Requisitos Funcionales	64
<i>Funcionalidad 1. Autenticación de usuarios</i>	64
<i>REQ-1: Deben Existir Dos Roles de Administrador y Usuario</i>	64

Funcionalidad 2. Generar solicitudes desde Usuario.....	64
REQ-1: Solicitud de Mantenimiento	65
REQ-2: Tipos de Solicitudes	65
Funcionalidad 3.....	65
Funcionalidad 4.....	66
Funcionalidad 5.....	68
Requisitos No Funcionales	74
Eficiencia	74
Seguridad lógica y de datos.....	75
Usabilidad.....	75
Análisis.....	75
Análisis del proceso de negocios.....	75
Desarrollar casos de uso.....	76
Investigación de la Oferta.....	77
Servidores Web.....	77
Bases de Datos	78
Frameworks o Librerías para Maquetado Frontend	78
Servicios en la Nube.....	79
Evaluación de la oferta	80
NodeJS: Servidor Web	80
MongoDB Compass: Base de Datos.	81
Elasticsearch: Motor de Búsqueda y Base de Datos	81
ReactJS: Librería para Maquetado Frontend	81
Arquitectura de servicios	81
Arquitectura del Frontend.....	82

Arquitectura del Backend	83
<i>Ingeniería de procesos de negocio</i>	83
<i>Documentación tradicional</i>	83
<i>Diseñar casos de prueba</i>	85
<i>Diseño de Interfaz de Usuario</i>	87
Diseño gráfico conceptual y experiencia de usuario	87
Diseño gráfico digital. Los diseños se realizaron uti.....	87
Maqueta.....	88
Implementación.....	89
<i>Desarrollo de servicio de negocio</i>	89
<i>Integración con la plataforma</i>	92
<i>Implementación</i>	94
Base de Datos.....	94
Clonación del repositorio.....	95
Instalación de dependencias.....	95
Inicio del servidor.....	95
Creación de zona alojada.....	96
<i>Pruebas Unitarias</i>	97
Aplicación Web.....	97
Microservicios	99
Pruebas de integración	100
Pruebas de rendimiento.....	101
Capitulo IV: Conclusiones y Recomendaciones	103
Conclusiones.....	103
Recomendaciones.....	104

Glosario de Términos.....	105
Bibliografía	106
Anexos	110

ÍNDICE DE TABLAS

Tabla 1 <i>Comparación entre Base de datos Relacional y Base de datos No Relacional</i>	45
Tabla 2 <i>Metodología SCRUM</i>	54
Tabla 3 <i>Caso de estudio Administrador</i>	76
Tabla 4 <i>Caso de uso Usuario</i>	76
Tabla 5 <i>Comparativa de ofertas de servidores web.</i>	77
Tabla 6 <i>Comparativa de ofertas en base de datos.</i>	78
Tabla 7 <i>Contraposición de ofertas en librerías para maquetado FrontEnd.</i>	79
Tabla 8 <i>Contraposición de ofertas en servicios en la nube.</i>	80
Tabla 9 <i>Casos de prueba</i>	85
Tabla 10 <i>El resultado de cargar la aplicación y cargar los componentes en el navegador</i>	97
Tabla 11 <i>Obtención resultados con tiempo de consulta optimizado en microservicios.</i>	99
Tabla 12 <i>Resultados de las pruebas de integración de microservicios y aplicaciones web</i>	100
Tabla 13 <i>Resultados de la prueba de rendimiento</i>	102

INDICE DE FIGURAS

Figura 1 <i>Estructura de una base de datos</i>	44
Figura 2 <i>Patrón cliente-servidor</i>	48
Figura 3 <i>Representación de las capas</i>	49
Figura 4 <i>Modelo vista controlador (MVC)</i>	50
Figura 5 <i>Patrón broker</i>	51
Figura 6 <i>Encuesta pregunta 1</i>	56
Figura 7 <i>Encuesta pregunta 2</i>	56
Figura 8 <i>Encuesta pregunta 3</i>	57
Figura 9 <i>Encuesta pregunta 4</i>	57
Figura 10 <i>Encuesta pregunta 5</i>	58
Figura 11 <i>Encuesta pregunta 6</i>	58
Figura 12 <i>Encuesta pregunta 7</i>	59
Figura 13 <i>Encuesta pregunta 8</i>	59
Figura 14 <i>Encuesta pregunta 9</i>	60
Figura 15 <i>Arquitectura primordial de una aplicación web</i>	82
Figura 16 <i>Correspondencia de la aplicación web entre el backend y el usuario</i>	82
Figura 17 <i>Manual de Usuario</i>	84
Figura 18 <i>Desarrollo del diseño de la interfaz Inicio de sesión con Moqups</i>	87
Figura 19 <i>Diseño de interfaz de Home Administrador</i>	88
Figura 20 <i>Diseño de Solicitudes</i>	88
Figura 21 <i>Progreso del diseño de interfaz con Moqups</i>	89
Figura 22 <i>Maquetado de bocetos Inicio de Sesión</i>	89
Figura 23 <i>Maquetado de Interfaz de Solicitudes</i>	90

Figura 24 <i>Codificación utilizando CSS</i>	90
Figura 25 <i>Maquetación mediante JSX</i>	91
Figura 26 <i>Maquetado de Home CSS</i>	91
Figura 27 <i>Validación de API</i>	92
Figura 28 <i>Codificación de las correspondencias o conexiones</i>	93
Figura 29 <i>Validación mediante API</i>	93
Figura 30 <i>Remisión de nuevo usuario</i>	94
Figura 31 <i>Selección del proveedor de Cloud y ubicación del clúster para MongoDB</i>	94
Figura 32 <i>Clonación del repositorio en la instancia</i>	95
Figura 33 <i>Instalación de las dependencias en el servidor</i>	95
Figura 34 <i>Inicio del servidor Amazon Web Services</i>	96
Figura 35 <i>Creación de zona alojada en el servidor</i>	96

Resumen

En la actualidad los sistemas de información son considerados como uno de los recursos más valiosos para cualquier tipo de organización, así mismo los avances tecnológicos en equipamiento en especial los que tienen que ver con el manejo de la información que han facilitado de forma significativa la labor de las empresas e instituciones, generando el desarrollo de las organizaciones, por ello es que se invierten considerables recursos económicos y grandes esfuerzos para implementar un respectivo plan de mantenimiento preventivo y correctivo con el fin de asegurar la disponibilidad inmediata de la información y garantizar la protección de los equipos informáticos. Por tales motivos y dando énfasis a una planificación adecuada de los recursos tecnológicos y a la importancia de la información, resulta prioritario y es obligación el establecer planes de mantenimiento preventivos y correctivos que nos permita como área de soporte a usuarios y mantenimiento técnico de la Radio Stereo Latacunga para prevenir de la mejor forma posible situaciones y vulnerabilidades en los equipos de la información y comunicación (TIC) que atenten contra la continuidad de los mismos y que afecten a la información contenida. Por consiguiente el departamento técnico de la Radio, a través del área de soporte a usuarios y mantenimiento técnico tiene la necesidad de establecer planes de mantenimientos preventivos y correctivos para evitar daños en los equipos, encontrar y corregir los problemas menores antes que estos provoquen fallas y grandes consecuencias a futuro, toda esta actividad ejecutarla con el propósito de proteger y mantener en funcionamiento los equipos de la información y comunicación (TIC) y así asegurar el procesamiento, la continuidad y la optimización de los recursos tecnológicos de la Institución.

Palabras clave. Radio stereo Latacunga, WEB, mantenimiento preventivo y correctivo, JAVA, medio de comunicación

Abstract

At present, information systems are considered one of the most valuable resources for any type of organization, as well as technological advances in equipment, especially those that have to do with information management, which have significantly facilitated the work of companies and institutions, revealing the development of organizations, which is why considerable economic resources and great efforts are invested to implement a respective preventive and corrective maintenance plan in order to ensure the immediate availability of information and guarantee the protection of computer equipment. For these reasons and emphasizing adequate planning of technological resources and the importance of information, it is a priority and an obligation to establish preventive and corrective maintenance plans that allow us as a support area for users and technical maintenance of Radio Stereo. Latacunga to prevent in the best possible way situations and vulnerabilities in information and communication equipment (ICT) that threaten their continuity and that affect the information contained. Therefore, the technical department of the Radio, through the area of user support and technical maintenance, has the need to establish preventive and corrective maintenance plans to avoid damage to the equipment, find and correct minor problems before they cause failures. and great consequences for the future, all this activity will be carried out with the purpose of protecting and maintaining the information and communication equipment (ICT) in operation and thus ensuring the processing, continuity and optimization of the Institution's technological resources.

Keyword. The radio stereo Latacunga, responsive web desing, preventive and corrective maintenance, dynamic content, scrum methodology

Capítulo I

Introducción

Antecedentes

Hoy en día, los sistemas informáticos se utilizan en todas las actividades que realizamos en todo el mundo, ya sea en el trabajo o en el juego. En el campo de la comunicación, el uso de herramientas informáticas se ha convertido en una prioridad en el campo de la gestión y gestión del mantenimiento preventivo y correctivo en diferentes campos de los medios de comunicación, lo que nos permite obtener múltiples beneficios en el campo de la comunicación, telecomunicaciones y TIC, así como estéreo radios e instituciones, para uso personal o uso de información a nivel institucional. En este caso desarrollaremos un sistema de gestión para controlar el mantenimiento preventivo y correctivo de equipos TIC (SGMPC), que permita registrar y controlar el mantenimiento realizado en los equipos de cómputo de "Radio Estéreo Latacunga". Cuando se diseñan y aplican correctamente, los sistemas informáticos nos brindan múltiples beneficios, permitiéndonos optimizar recursos, reducir el tiempo de desarrollo de procesos y brindarnos información a la hora de tomar decisiones.

Por la trascendencia se han desarrollado trabajos como los que se detallan a continuación:

- Software UpKeep - UpKeep Technologies / Ubicado en Los Ángeles, EE. UU / 2014. Del cual es la solución moderna de gestión de activos y CMMS para su equipo. Desde su escritorio hasta su teléfono o incluso una tableta, es fácil acceder a UpKeep en cualquier lugar y en cualquier momento. Cree órdenes de trabajo sobre la marcha, reciba notificaciones cuando se actualicen las tareas y reciba alertas directamente

desde su aplicación cuando los activos estén bajos para que su negocio funcione de manera más eficiente que nunca (UpKeep, 2023).

- FIIX - Rockwell Automation - Ubicado en Toronto, Canadá – 2008. Planifique, organice y realice un seguimiento del mantenimiento preventivo con Fiix, un CMMS basado en la nube que le permite beneficiarse de un mantenimiento ilimitado. Los tableros personales de Fiix permiten a los gerentes acceder a la información y mejorar los gerentes de programas, mientras que una experiencia móvil sin precedentes permite a los técnicos mantener a los gerentes actualizados con información desde el piso de la tienda o en la escena (Fiix, 2023).

Por lo expuesto es fundamental que la Radio Latacunga FM-AM disponga de un sistema informático que le permita sistematizar y controlar los procesos de mantenimiento preventivo y correctivo en cada una de las áreas de las que este dispone.

Planteamiento del Problema

La Radio Latacunga FM-AM está ubicado en el cantón Latacunga, proporciona el entretenimiento e información que la gente necesita según su demografía, geografía, etnografía, información verídica con calidad y calidez, en apoyo de la sociedad, mediante la evaluación, transmisión, a fin de disponer de información apto para toda la sociedad, como aporte a los intereses institucionales y del Estado; y con su capacidad a disposición de equipos tecnológicos en excelente estado.

En la actualidad la Radio Latacunga FM-AM se encuentra sin utilizar un software de gestión de mantenimiento preventivo y correctivo que permita la gestión y control de sus procesos, por lo cual este software será implementado.

Justificación

La Radio Latacunga FM-AM se caracteriza por devolver al pueblo la voz que fue acallada, por lo tanto no solo es la voz oficial de la Diócesis de Latacunga, sino es la voz de un pueblo en marcha, entendida a la diócesis como parte del pueblo que camina, se busca mantener una adecuada gestión de mantenimiento preventivos y correctivos de cada uno de equipos TIC's, se pretende desarrollar una aplicación web basado en la metodología MVC (Modelo Vista-Controlador) siendo una metodología ágil y adecuada para el desarrollo de software, teniendo como objetivo satisfacer las necesidades de la radio.

Actualmente, las aplicaciones web son usadas en casi todos los medios de comunicación ya que estas permiten llevar un correcto control, gestión y administración de los equipos TIC's y herramientas que disponen las mismas, otros aspectos importantes son:

- Permite que se pueda acceder desde cualquier lugar, equipo y dispositivo independientemente del sistema operativo que manejen los mismos.
- Las aplicaciones web disponen de una interfaz gráfica muy amigable y fácil de usar para los usuarios.
- Permite realizar búsquedas de manera fácil y obtener resultados de manera inmediata.
- Toda la información que maneja el sistema es alojada en la base de datos, de esta manera se mantiene la confidencialidad, integridad y disponibilidad.

La aplicación web tiene como objetivo gestionar los mantenimientos de equipos TIC's, materiales y herramientas de la radio.

Se beneficiarán de forma directa el administrador de la radio y los usuarios pertenecientes a las diferentes dependencias del mismo, además servirá de ejemplo para futuros proyectos.

Por lo expuesto es importante que la Radio Latacunga FM-AM disponga de una aplicación web que sirva para gestionar los mantenimientos preventivos y correctivos de

equipos TIC's, materiales y herramientas de la Radio con lo cual será de gran ayuda para el administrador de la Radio.

Objetivos

Objetivo General

Construir un aplicativo Web que permita ejecutar un plan de mantenimiento preventivo y correctivo de software y hardware en los equipos TIC'S en la radio stereo Latacunga ubicado en el cantón Latacunga.

Objetivos Específicos

- Investigar el estado del arte de las herramientas actuales y aplicativos web que permitan construir un plan de mantenimiento.
- Recolectar datos de los equipos TICS actuales en la radio.
- Desarrollar la metodología de Software enfocado a un plan de mantenimiento de equipos TIC'S.
- Implementar el aplicativo web en el servidor de la radio.

Alcance

El presente proyecto tiene como finalidad principal realizar la entrega a la Radio Latacunga FM-AM una aplicación web.

El desarrollo de la aplicación web para la Radio Latacunga FM-AM permitirá mejorar la gestión de los mantenimientos preventivos y correctivos de equipos TIC's; dado que se conocerá la ubicación física, el inventario, el responsable del mantenimiento de dichos equipos y la dependencia a la que fueron asignados.

El registro de mantenimientos preventivos y correctivos de equipos TIC's se lo realizara de manera asíncrona con la base de datos MongoDBCompass, permitiendo mantener actualizado el registro de mantenimientos luego de cualquier realización que exista en la misma.

El sistema permitirá dar de baja a los equipos que estén en defectuoso estado, con la finalidad de mantener un control de acceso a la aplicación manteniendo de esta manera la integridad de la información del sistema.

Al finalizar el proyecto se busca que los usuarios puedan acceder desde cualquier lugar y dispositivo a la aplicación web para conocer el estado de los equipos TIC's, mantenimientos preventivos y correctivos que se ejecutaron en la Radio Latacunga FM-AM.

Capítulo II

Marco Teórico

Software

El software es un conjunto de componentes lógicos intangibles, pero es muy necesario para realizar una tarea específica en nuestro sistema, vale la pena mencionar que el software está diseñado, el software no fallará y el software se puede personalizar de acuerdo con los requisitos del cliente. (Moreno & Ramos, 2014).

Para que los programas funcionen correctamente, necesitan métodos que les permitan funcionar, como estructuras de datos que permitan manipularlos y documentos que describan sus operaciones. El término software se utiliza para referirse a toda la colección, comenzando con un sistema informático que consta de programas, procedimientos, reglas informáticas, documentos y datos relacionados necesarios para realizar determinadas tareas. El término software apareció en oposición al término hardware, que se usaba para nombrar los componentes físicos de una computadora en un intento de indicar su intangibilidad (Brocca & Camasaqui, 2005, pág. 6).

Licencias de Software

En general, todo el software se basa en una licencia que define todas las políticas que limitan el uso y la distribución del software. El incumplimiento de estas políticas puede dar lugar a que el creador emprenda acciones legales contra los usuarios finales que no cumplan con la licencia bajo la cual se divulgó el software. En política.

Al instalar, utilizar o copiar el producto de software con licencia, usted acepta estar obligado por los términos del acuerdo de licencia. Por lo tanto, es importante leer detenidamente la licencia del programa para comprender sus términos de uso. Cabe señalar que algunas licencias son tan restrictivas que incluso las realizadas con las herramientas

adecuadas no se pueden obtener de forma gratuita, mientras que otras requieren modificación o reproducción (Brocca & Camasaqui, 2005, pág. 4).

Categorías De Software

El software se divide en tres amplias categorías según las limitaciones del uso de cada software para el usuario final.

Software Libre

Es un software que respeta la libertad de los usuarios y las comunidades. En términos generales, esto significa que los usuarios son libres de ejecutar, copiar, distribuir, investigar, modificar y mejorar el software, lo que les permite controlar el programa y sus funciones. Para ser libre, el software debe respetar las 3 libertades básicas: ejecutar, probar y distribuir (Arteaga , 2019).

Software Open Source

Originalmente, el término código abierto (o código abierto) significaba software de código abierto (OSS). El software de código abierto es un código diseñado para estar disponible públicamente: cualquiera puede ver, modificar y distribuir el código como mejor le parezca. El software de código abierto se desarrolla de manera descentralizada y colaborativa, basándose en la revisión por pares y la producción comunitaria. Además, a menudo es más accesible, flexible y sólido que sus alternativas propietarias porque la comunidad es responsable de su desarrollo, en lugar de un solo autor o empresa (RedHat, 2020).

Software Propietario

Es cualquier programa informático que restringe a los usuarios su uso, modificación o distribución, incluso si su código fuente no está disponible, pero es uno de los programas más famosos que impide que los usuarios lo modifiquen (EAE, 2020).

Debido a que el software propietario representa esencialmente los intereses comerciales del desarrollador, se debe obtener una licencia para su uso legal. Por el mismo

motivo, periódicamente se lanzan al mercado nuevas versiones del producto para corregir los errores detectados e incluir nuevas funcionalidades. Son estas nuevas demostraciones las que tarde o temprano obligarán a los usuarios a migrar a una nueva versión actualizada con costos asociados. También debe tenerse en cuenta que después de un cierto período, la empresa comercializadora deja de brindar soporte técnico para el producto. También es una razón común para la migración (Brocca & Camasaqui, 2005).

Dominios de Aplicación del Software

Debido a la aparición reciente de una amplia variedad de software, se clasifica según los dominios a los que se dirige, lo que permite identificar siete dominios de aplicación de software:

Software de Sistemas

Proporciona un alto grado de interacción con el hardware de la computadora, uso intensivo por parte de muchos usuarios, operaciones simultáneas que requieren secuenciación, recursos compartidos y control de procesos complejos. Estructuras de datos complejas y múltiples interfaces externas (compiladores, editores de texto, software de red, componentes del sistema operativo) (Pressman, 2010).

Software de Aplicación

Estos son programas individuales que abordan necesidades comerciales específicas. Las aplicaciones en esta área procesan datos comerciales o técnicos de una manera que facilita la toma de decisiones comerciales, administrativas o técnicas. Además de las aplicaciones normales de procesamiento de datos, este software se utiliza para controlar las funciones comerciales en tiempo real (Pressman, 2010).

Software de Ingeniería y Ciencias

La característica principal es que consta de algoritmos de detección de números. Sin embargo, las aplicaciones modernas en ingeniería y ciencia abandonan los algoritmos

numéricos tradicionales. Las aplicaciones interactivas, como el diseño asistido por computadora y la simulación de sistemas, comienzan a funcionar en tiempo real e incluso tienen las características del software del sistema (Pressman, 2010).

Software Incrustado

Existe en un producto o sistema para implementar y controlar funciones y características de usuarios finales y sistemas. La funcionalidad del software es limitada y específica, o para proporcionar una funcionalidad y un control significativos (Pressman, 2010).

Software de Línea de Productos

El software atiende específicamente a diferentes consumidores, el software se puede usar en un mercado específico o se puede usar en un mercado de consumo masivo donde se procesan grandes volúmenes de datos (Pressman, 2010).

Software de Inteligencia Artificial

Utiliza innumerables algoritmos para resolver problemas complejos que no pueden manejarse fácilmente de manera computacional o directamente analítica. Las aplicaciones en esta área incluyen robótica, sistemas expertos, reconocimiento de patrones (imagen y voz), redes neutrales artificiales, demostración de teoremas y juegos (Pressman, 2010).

Aplicaciones Web

Estas aplicaciones también saben en nombre de WebApps. En el formulario más simple, las aplicaciones web son solo enlaces a archivos de texto Super. Estos archivos muestran información con texto y gráficos limitados. Sin embargo, Web 2.0 está desarrollando un entorno informático sofisticado. Este entorno informático no solo proporciona a los usuarios propiedades de aislamiento, computadoras y funciones de contenido, sino que también se integra con bases de datos de la empresa y aplicaciones comerciales. (Pressman, 2010).

Tipos de Mantenimientos de un equipo informático

Mantenimiento

Mantenimiento es el nombre que se le da al procedimiento mediante el cual se trata un determinado bien para que no se vea afectado por el paso del tiempo, el uso o los cambios en el medio exterior (Definición de Mantenimiento - Qué Es Y Concepto, 2014).

Mantenimiento preventivo. Como sugiere el nombre, se trata de acciones que se toman para evitar errores y mantener el hardware de la computadora en buen estado de funcionamiento. El mantenimiento informático preventivo se puede definir como una serie de trabajos realizados por profesionales para optimizar la máquina. El objetivo es reducir las fallas de la computadora y mejorar el rendimiento. Para lograr este objetivo, el software se mantiene para garantizar que los programas, como el software antivirus, alcancen los límites de rendimiento previstos. También tratamos el hardware, la parte "física" de nuestra computadora, limpiando y alineando los componentes (Mantenimiento Preventivo de Computadores Colombia - SOTEIN, 2022).

Cabe recalcar que el mantenimiento preventivo debe realizarse periódicamente de acuerdo al uso de los equipos de cómputo. Si el nivel de uso es bajo, se puede realizar un mantenimiento preventivo una vez al año. Si el uso es moderado, entonces cada 6 meses límpielo una vez al mes (Mantenimiento Preventivo de Computadores Colombia - SOTEIN, 2022).

Mantenimiento correctivo. Se conoce como una operación de computadora que soluciona problemas de computadoras portátiles, servidores, PC y varios dispositivos de hardware. El mantenimiento correctivo informático es un programa de acciones planificadas para eliminar errores en la infraestructura informática lo más rápido posible (Mantenimiento Correctivo de Computadoras, Ordenadores O PC, 2018).

Mantenimiento correctivo de hardware. En este caso, debe reemplazar un componente de su computadora, ya sea una computadora portátil, una terminal minorista o cualquier tipo de computadora. El mantenimiento correctivo del hardware es responsable de diagnosticar fallas, determinar qué ha sucedido y reemplazar el hardware dañado. Es importante que los técnicos tengan experiencia, materiales y puedan llegar rápido. Cuando un técnico llega a su oficina o sede, puede volver al trabajo en unas pocas horas (Mantenimiento Correctivo de Computadoras, Ordenadores O PC, 2018).

Mantenimiento correctivo de software. En cuanto a los programas informáticos como ERP, CRM y contabilidad, y los componentes informáticos de las aplicaciones comerciales, se pueden realizar diversas actividades. El mantenimiento correctivo de procedimientos se utiliza para corregir errores que ocurren en el software, incluidas las aplicaciones de oficina como Office, Excel, Access, Word y aplicaciones comerciales para administración, nómina, contabilidad o gestión de inventario (Mantenimiento Correctivo de Computadoras, Ordenadores O PC, 2018).

Este tipo de mantenimiento también se encarga de corregir errores en el sistema operativo que impiden que arranque. Se pueden encontrar problemas de virus o se pueden eliminar archivos importantes. Incluso puede hacer que el sistema operativo quede inutilizable. La única forma de guardar su computadora para usarla es formatear el disco duro (Mantenimiento Correctivo de Computadoras, Ordenadores O PC, 2018).

Lenguajes de programación

Es una herramienta que te permite crear programas en un lenguaje que las computadoras puedan entender para procesar datos, y es una parte esencial del diseño que te permite crear algoritmos o completar cálculos (Vértice, 2009).

Lenguaje de Merca de Hipertexto (HTML)

Significa Hypertext Markup Language, o también llamado Hypertext Markup Language, que es el lenguaje básico de hipertexto que contiene referencias como enlaces, enlaces a otro texto. Esto significa que nos brinda la mayor ventaja de compatibilidad en tiempo de diseño y facilidad de uso, ya que nos brinda las herramientas que necesitamos para crear documentos basados en el almacenamiento del servidor web. Los programadores son responsables de editar y crear texto, guardando archivos con extensión de archivo *.html. La sintaxis son los pasos a seguir al crear un programa, como etiquetas, atributos, elementos (Vértice, 2009).

Hojas de Estilo en Cascada (CSS)

Se define como Cascading Style Sheets (CSS), lo que nos permite mejorar el diseño de la interfaz en cuanto a diseño, ubicación y apariencia. Separar y organizar el contenido. Ser capaz de administrar bien los datos aumentará la velocidad de las páginas web y ahorrará líneas de código. En 1994, el científico Håkon Wium Liel fue el impulsor de la mejora de la red a través de mejoras individuales y en cascada. W3C (Rubiales Gómez Mario, 2021).

JavaScript

El objetivo de este lenguaje de programación es crear páginas web dinámicas que puedan interpretarse y ejecutarse sin compilación en un programa. Todo comenzó en la década de 1990, Internet estaba en auge y la gente buscaba nuevas tecnologías para diseñar mejor las páginas web, hacerlas más sencillas y mejores, un lenguaje que permite ejecutar programas sin iniciar el navegador del usuario fue Breandar Eich. un hombre que resolvió los grandes problemas de su época con tecnologías como ScriptEase, que introdujo en 1995. Originalmente llamó a su lenguaje LiveScript. Netscape firmó la alianza por mixrosystem y decidió cambiar el nombre a JavaScript (Eguíluz Pérez Javier, s. f.).

NodeJS

Ryan Dahl lanzó la primera versión de node.js en 2009, que desarrolló el concepto de Javascript independiente del servidor. Es un entorno para ejecutar aplicaciones utilizando el lenguaje de programación Javascript. La principal ventaja es que su código fuente es abierto y cualquiera puede acceder a él e instalarlo en nuestro ordenador sin pagar licencia (Puciarelli, 2020).

ReactJS

Sigue siendo de código abierto como una biblioteca de Javascript que permite crear interfaces de usuario para facilitar el desarrollo de aplicaciones. React.js se caracteriza por su sencillez y facilidad de uso. Necesitamos asegurarnos de que solo maneje la parte frontal de la aplicación usando el patrón MVC, que significa MODEL-VIEW-CONTROL. La migración de versiones es muy fácil de manejar (Anchundia Medrano, 2022).

JSX

Es una extensión de JavaScript utilizada por React que nos permite agregar lenguajes como HTML para ejecutar componentes. Interpreta el código y lo compila en formato JSX y lo traduce a JS para enviar la solicitud al navegador. La principal ventaja es que reduce el código de desarrollo de aplicaciones (Jiménez Calderón, Mario, 2020).

Servidor web y servicios en la nube

Para entender en qué consiste un servidor web, es necesario definir los siguientes términos que abarcan su funcionamiento.

Servicio Web

Es un sistema integral de acceso a la información multimedia que permite a los terminales conectados a dicha red solicitar información. El sistema fue desarrollado originalmente por el CERN (Centre Européen de Recherche Nucléaire) en 1990 y fue implementado para mejorar el flujo de información que la entidad necesita para sus

investigaciones. En 1992, se lanzó al público el primer navegador web llamado "Mosaic" (Cobo, Gómez, Pérez, & Rocha, 2005).

Modelo Cliente Servidor

El modelo define claramente el comportamiento del servidor web utilizado, basado en dos computadoras conectadas en la misma red, donde se dividen los roles de cliente y servidor (Cobo, Gómez, Pérez, & Rocha, 2005).

Servidor

Se trata de dispositivos informáticos con grandes recursos que pueden dar servicio a los demás dispositivos conectados a la misma red. Dado que es un proveedor de servicios, su dirección IP debe ser estática para evitar problemas de conexión en el futuro. Estas computadoras alojan páginas web, aplicaciones móviles, sistemas de administración de bases de datos y otros servicios (Cobo, Gómez, Pérez, & Rocha, 2005).

Cliente

Es un dispositivo informático utilizado por un usuario individual. Siempre que esté en la misma red, puede enviar una solicitud de información al servidor. Por lo tanto, no es necesario mantener una IP estática, ya que solo envía solicitudes y no envía solicitudes (Cobo, Gómez, Pérez, & Rocha, 2005).

Servidor Node.js

Es un entorno de tiempo de ejecución JavaScript asíncrono basado en eventos para desarrollar aplicaciones web escalables. En el código de Node.js, se realizan muchas conexiones simultáneas que mantienen el sistema en funcionamiento hasta que deja de aceptar devoluciones de llamada, es decir, Node.js se detiene. Además, te permite crear un servidor HTTP diseñado para una mejor transferencia de datos a través de streaming y baja latencia. Por lo tanto, se puede usar fácilmente en aplicaciones web que funcionan con marcos web (OpenJS Foundation, 2021).

Amazon Web Services

Es un proveedor de servicios en la nube multitecnología que aloja todo el servicio en servidores flexibles que se adaptan a las necesidades comerciales, y sus productos van desde tecnologías de infraestructura como cómputo, almacenamiento y bases de datos hasta tecnologías emergentes como aprendizaje automático e inteligencia artificial, lagos de datos y Analítica e IoT. Hace que sea más rápido, fácil y rentable poner las aplicaciones existentes en la nube y le permite crear casi cualquier servicio o incluso producto que requiera un software especial. (Amazon Web Services, Inc., 2021).

Metodología de Desarrollo de Aplicaciones

El proceso de desarrollo técnico necesita incluir estrategias y métodos de trabajo que mejoren el proceso y tiempo de entrega de los proyectos de software. Por lo tanto, es una parte esencial del desarrollo de cualquier producto o proyecto de software.

Metodología Scrum

Antecedentes. En el pasado, el método en cascada se usaba ampliamente en el desarrollo de aplicaciones de software, pero debido a la necesidad actual de un desarrollo más rápido, este método de trabajo se ha vuelto obsoleto. Desde 1990, el método SCRUM surgió y ganó popularidad con el tiempo debido a su gran capacidad para entregar proyectos de software en un tiempo relativamente corto (Dimes, 2015).

Definición. Además de la metodología, es un marco de referencia para el uso de métodos ágiles de desarrollo de software. Cómo se puede usar este enfoque para entregar un gran producto de manera acelerada, pero uno de sus principios básicos es el uso de una "pila de productos" que tiene como objetivo alinearse con las características que hacen un gran producto y priorizar su desarrollo. Por encima de todo lo anterior (Dimes, 2015).

Desarrollo de Software Agile. Este es un enfoque centrado en el ser humano que reduce el uso de documentación innecesaria y proporciona un producto más preciso basado en las acciones y recomendaciones del cliente final que satisfacen las necesidades reales. Este es el factor diferenciador, ya que los enfoques tradicionales solo intentan generar investigación previa y desconocen la contribución de todo el ciclo de vida del proyecto (Dimes, 2015).

Perfiles de la Metodología Scrum

Dado que este es un método de trabajo, es necesario definir el equipo responsable del desarrollo de la aplicación, cuyo costo se determina en el flujo de trabajo.

Product Owner. Es responsable de definir los objetivos del proyecto en función de los aportes del cliente final y, en función de estos, puede definir de manera clara y precisa los objetivos que el equipo de desarrollo debe alcanzar.

Scrum Master. Es el líder del equipo de trabajo, porque es el responsable de la motivación del equipo y de las ganas de alcanzar las metas que se propongan. Sirve como motor de proyectos y como guía en momentos difíciles.

Scrum Team. Es el equipo responsable del desarrollo y entrega del producto final, una estructura horizontal autoorganizada con el objetivo de poder definir roles y tareas de acuerdo a las habilidades de los miembros del equipo.

Stakeholder. Pueden intervenir personas ajenas al equipo por tratarse de perfiles interesados en el producto, como el mismo cliente o responsable de la empresa que solicitó el desarrollo del producto.

Etapas de la Metodología Scrum

Las fases de la metodología Scrum mantienen un marco de inicio y un proceso de bucle que se genera después de que comienza la fase de desarrollo.

Etapas 1: Product Backlog. El proceso comienza con la elaboración del denominado Product Backlog. Es un documento general que incluye un conjunto de tareas, requisitos y capacidades para un proyecto. Cualquier miembro del equipo puede cambiar este documento, pero la única persona autorizada para agregar una prioridad es el propietario del producto responsable del documento. El autor Mike Kohn, experto en empresas de software, utiliza el acrónimo "DEEP" para indicar las distintas etapas o fases de la creación de un buen producto. Corresponden a:

Detailed Appropriately: En esta fase, los requisitos del producto se definen como las características que afectan el desarrollo del producto.

Emergent: Esta parte define el Product Backlog como algo que nunca deja de crecer y cambiar ya que siempre se adapta a las necesidades del cliente y por lo tanto a las decisiones del Product Owner.

Estimated: esta parte significa "estimado" porque se refiere a una estimación aproximada basada en el esfuerzo y el valor asignado al proyecto.

Prioritized: esta parte define el hecho de que todos los elementos en la cartera de productos deben priorizarse y categorizarse.

Etapas 2: Sprint Backlog

Es un documento que especifica las tareas a realizar y quién debe realizarlas. Es divertido desglosar las horas de trabajo requeridas para cada uno y aplicarlas al costo. Si su volumen es alto, la creación de hitos ayudará. Un Sprint es un período en el que se llevan a cabo todas las actividades acordadas en el Sprint Backlog, lo que significa entregables parciales para probar el producto final. Se debe repetir el ciclo anterior hasta entregar todos los elementos Blacklog. No debe haber tiempo improductivo entre sprints.

Etapas 3: Sprint Review

Todo lo que hacemos debe ser controlado. En Burn Down destacamos los estados y su evolución, especificando las tareas y requisitos a mapear. Al final del sprint, se revisa todo el trabajo, lo cual es una buena oportunidad para obtener comentarios sobre el desarrollo del producto. Puede ser una reunión informal siempre que tengas claro el propósito de la Revisión de Sprint: brindar transparencia al equipo y a los clientes. Un sprint de cuatro semanas suele durar unas cuatro horas. Eso significa una hora a la semana. El responsable de realizar este trabajo es el Product Owner, mientras que el Scrum Master se asegura de que así sea y de que se cumplan los plazos. Entonces, ¿qué está pasando en Sprint Review? En esta etapa, ya sabemos que el propietario del producto es el único responsable de la revisión, es el responsable de explicar los elementos atrasados y completarlos.

El equipo de desarrollo, por otro lado, es responsable de demostrar y responder preguntas sobre el progreso realizado durante el sprint. Luego se revisa el proyecto y se discuten los próximos pasos. Si el cliente recibe una respuesta, el Dueño del Producto reordenará el saldo de los Productos. Finalmente, se realiza una revisión del tiempo, presupuesto y alcance del proyecto final.

Etapas 4: Sprint Planning Meeting

¿Quién no ha desperdiciado horas de trabajo inútil en reuniones improductivas por una preparación inadecuada? No tiene cabida en los métodos ágiles. Cada minuto cuesta dinero. Las reuniones también deben programarse como parte del proceso. En esta "Reunión de Planificación de Sprint", el Product Owner prioriza las tareas contenidas en el Product Backlog. Con estas tareas en mente, establezca objetivos para el nuevo sprint, priorice las tareas que debe completar el equipo Scrum y asigne tiempo para cada tarea. Los objetivos deben ser alcanzables y el equipo solo completará un conjunto de tareas alcanzables. Esta reunión monitorea el proyecto diariamente y verifica la finalización de las tareas asignadas durante la

reunión. Es posible que haya oído hablar de Daily Scrum, su nombre está tomado del inglés. En esta reunión se acuerdan las metas para el día siguiente y se analizan los temas que pueden limitar o dificultar directamente el logro de las metas (GALIANA, 2021).

Ciclo de vida de un proyecto de Software

Todos los productos pasan por un proceso, analizan el proceso y ven los resultados, por lo que el desarrollo de proyectos de software no está obsoleto, requiere un proceso antes de que el producto pueda exponerse a los clientes. Existen diferentes modelos de estos, entre los que tenemos el modelo en cascada, el modelo en espiral, el de desarrollo gradual, etc. Todos estos métodos deben satisfacer al cliente, por lo que todos deben cumplir con los objetivos establecidos por el cliente final (JAVIER MARTÍNEZ, 2013).

Recopilación de los Requerimientos

Recopilar y dialogar sobre las características clave de las necesidades del cliente, las cuestiones organizativas, los objetivos estratégicos de la empresa y los departamentos de la empresa para revisar los requisitos de la empresa (JAVIER MARTÍNEZ, 2013).

Análisis

Recibir y comunicar las necesidades de los clientes a la organización mediante la recepción y clasificación de solicitudes, el envío de solicitudes. Esta parte establece el protocolo de comunicación entre el analista y el usuario final (JAVIER MARTÍNEZ, 2013).

Limitaciones

Decirle al cliente que determine la culminación de la aplicación para comenzar a definir la investigación y los recursos que tenemos (JAVIER MARTÍNEZ, 2013).

Especificación

Denota la funcionalidad del sistema y sus limitaciones de desarrollo (JAVIER MARTÍNEZ, 2013).

Diseño y Arquitectura

Obtener los requisitos arquitectónicos generales de la aplicación para prototipado. Los prototipos ayudarán a los clientes/usuarios a definir los requisitos del proyecto (JAVIER MARTÍNEZ, 2013).

Programación

Generar componentes de software que cumplan con la definición del producto y cumplan con los requisitos. La funcionalidad especificada se implementa utilizando el lenguaje de programación seleccionado. Crear ayuda en línea y guías de usuario. Después de diseñar el producto, se crea un sistema interactivo (JAVIER MARTÍNEZ, 2013).

Pruebas de software

Pruebe los componentes del software según el plan de calidad para verificar que el software cumpla con los requisitos. Ejecutar el plan de pruebas, especificar las pruebas a desarrollar, ejecutar y evaluar. reportar los resultados (JAVIER MARTÍNEZ, 2013).

Implementación

La implementación del software se planifica y ejecuta, asegurando la implementación del sistema de manera que garantice el cumplimiento de los estándares de calidad de la empresa, las expectativas del cliente y la seguridad e integridad de la información. Brindar soporte a los usuarios (JAVIER MARTÍNEZ, 2013).

Mantenimiento

Introduce un enfoque estructurado para ofrecer mecanismos de control para un plan de mejora del proceso de desarrollo de software empresarial, incluido el desarrollo de indicadores de eficiencia, eficacia y satisfacción para los clientes internos y externos de la empresa y mecanismos para el control y seguimiento del proceso de diseño (JAVIER MARTÍNEZ, 2013).

Motor de búsqueda

Es una herramienta para encontrar la información que el usuario necesita sobre cualquier tema, y la información reflejada por el motor de búsqueda se rastrea y se muestra con las etiquetas de palabras adecuadas. Un ejemplo obvio es Google (Pastor, 2016).

Elasticsearch

Es un motor de búsqueda y la mayor ventaja es que es gratuito y abierto y puede manejar todos los datos. Se basa en Apache Lucene lanzado por Elasticsearch N.V. en 2010. Ahora se llama Elástico. Incluye una API REST simple con gran escalabilidad. Se utiliza, por ejemplo, para monitorear aplicaciones, sitios web, contenedores, seguridad o análisis de negocios. Se necesitan los datos para interpretarlos y comprenderlos. Elasticsearch almacena todos los datos como documentos JSON. El procesamiento de datos es instantáneo (elastic, s. f.-a).

Monstache

Es un proceso Daemon llamado demonio en español que resuelve todas las solicitudes de forma sincrónica. Monstache se puede usar con Elasticsearch para realizar búsquedas complejas y agregaciones de datos de MongoDB y crear fácilmente visualizaciones y tableros de Kibana en tiempo real. (Ryan Wynn, 2016).

Bases de datos

Una base de datos es una estructura organizada de información o datos estructurados, generalmente reservados electrónicamente en un método informático. Por lo general, una base de datos es dirigida por un sistema de administración de bases de datos (DBMS). Los datos y el DBMS y las adaptaciones relacionadas se conocen colectivamente como el sistema de base de datos, a menudo denominado simplemente como la base de datos, existen dos tipos de base de datos (¿Qué Es Una Base de Datos?, 2014).

Base de datos Relacional

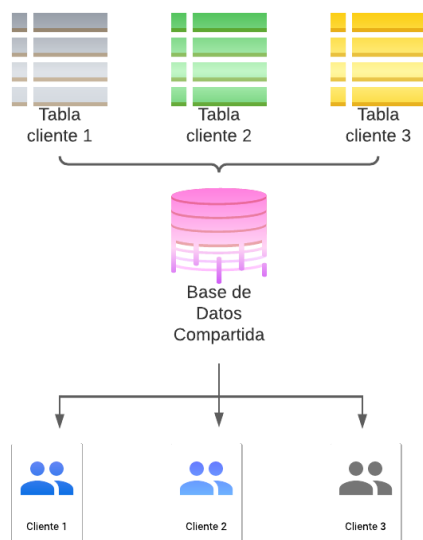
Una base de datos relacional es una base de datos que almacena y proporciona acceso a puntos de datos que están relacionados entre sí. Las bases de datos relacionales se basan en el modelo relacional, que es una forma intuitiva y sencilla de representar datos en tablas. En una base de datos relacional, cada fila de una tabla es un registro con un identificador único llamado clave. Las columnas de la tabla contienen atributos de datos y, por lo general, cada registro tiene un valor para cada atributo, lo que facilita la creación de relaciones entre puntos de datos (*¿Qué Es Una Base de Datos Relacional?*, 2014).

Estructura de las bases de datos relacionales

El modelo relacional significa que las estructuras de datos lógicos (tablas de datos, vistas e índices) están separadas de las estructuras de almacenamiento físico. Debido a esta separación, los administradores de bases de datos pueden administrar el almacenamiento físico de datos sin afectar el acceso a esos datos como estructura lógica. Por ejemplo, cambiar el nombre de un archivo de base de datos no significa que los nombres de sus tablas también cambien (*¿Qué Es Una Base de Datos Relacional?*, 2014).

La distinción entre lógico y físico también se aplica a las operaciones de bases de datos, que son operaciones bien definidas que permiten que las aplicaciones manipulen datos y estructuras de bases de datos. Las operaciones lógicas les dicen a las aplicaciones lo que necesitan, mientras que las operaciones físicas definen cómo se debe acceder a los datos y cómo se deben realizar las tareas (*¿Qué Es Una Base de Datos Relacional?*, 2014).

Para garantizar la precisión y disponibilidad continuas de los datos, las bases de datos relacionales siguen ciertas reglas de integridad. Una regla de integridad podría, por ejemplo, especificar que no se permiten filas duplicadas en una tabla para evitar que ingrese información errónea en la base de datos (*¿Qué Es Una Base de Datos Relacional?*, 2014).

Figura 1*Estructura de una base de datos*

Nota. Debemos tener en cuenta la estructura de una base de datos relacional.

Base de datos No Relacional

Las bases de datos NoSQL están trazadas para modelos de información específicos y tienen sinopsis flexibles para crear aplicaciones modernas. Las bases de datos NoSQL son extensamente reconocidas por su simplicidad de desenvolvimiento, operatividad y productividad. Esta página contiene recursos para ayudarlo a comprender mejor y comenzar a utilizar las bases de datos NoSQL (Bases de Datos No Relacionales | Bases de Datos de Gráficos | AWS, 2023).

Base de datos Relacional vs Base de datos No Relacional

Durante décadas, el modelo de datos dominante para el desarrollo de aplicaciones fue el modelo de datos relacionales utilizado por bases de datos relacionales como Oracle, DB2, SQL Server, MySQL y PostgreSQL. No fue sino hasta mediados o finales de la década de 2000

que se empezaron a adoptar otros modelos de datos y su uso aumentó significativamente. El término "NoSQL" se acuñó para distinguir y clasificar estas nuevas clases de bases de datos y modelos de datos. Los términos "NoSQL" y "no relacional" a menudo se usan indistintamente.

Aunque existen muchos tipos de bases de datos NoSQL con sus propias características, la siguiente tabla muestra algunas de las diferencias entre las bases de datos SQL y NoSQL.

Tabla 1

Comparación entre Base de datos Relacional y Base de datos No Relacional

	Base de datos relacional	Base de datos no relacional
Cargas de trabajo óptimas	Están diseñados para aplicaciones de procesamiento de transacciones en línea (OLTP) altamente consistentes.	Están diseñados para una variedad de modelos de acceso a datos, incluidas aplicaciones de baja latencia, están diseñadas para analizar datos semiestructurados.
Modelo de datos	La base de datos asegura la integridad referencial en las correspondencias entre tablas.	Proporcionan una variedad de modelos de datos que están optimizados para el rendimiento y la escala.
Propiedades ACID	Las bases de datos relacionales proporcionan propiedades de atomicidad, consistencia, aislamiento y durabilidad.	Estos se compensan adaptando algunas de las características ACID de las bases de datos relacionales para un modelo de datos más adaptable.
Rendimiento	El rendimiento óptimo requiere la optimización de las consultas, los índices y la estructura de la tabla.	Esto depende del tamaño del clúster de hardware implícito, la latencia de la red.
Escalado	Reducen la potencia informática del hardware o agregan réplicas a las cargas de trabajo de solo lectura.	Se pueden distribuir porque el modelo de acceso es escalable, lo que mejora el rendimiento con una arquitectura distribuida.

	Base de datos relacional	Base de datos no relacional
API	Solicitudes para almacenar y recuperar datos enviados por consultas compatibles con lenguaje de consulta estructurado (SQL).	Una API basada en objetos posibilita a los desarrolladores guardar y recuperar accesiblemente estructuras de datos.

Nota. Podemos distinguir la diferencia entre una base de datos relacional y una base de datos no relacional.

Modelos de SGBD

El modelo de datos consta de las siguientes partes:

- Conjunto de conceptos que definen la estructura de una base de datos:
 - Datos.
 - Relaciones entre datos.
 - Limitaciones de datos y relaciones.
- Un conjunto de operaciones para consultar y actualizar datos.

Según el tipo de concepto, el modelo de datos se divide en:

- Conceptual
- Logico
- Físico

Modelo conceptual

Esto es más relevante en la fase de concepto, cuando la empresa prepara los planes preliminares y los planes para desarrollar los detalles más tarde. Los modelos de datos conceptuales, normalmente creados por arquitectos de datos y partes interesadas del negocio, brindan a las partes interesadas una instantánea comprensible de los conceptos o entidades relacionados y las relaciones entre ellos (Modelado de Datos Conceptual | Erwin, Inc., 2023).

- Entidad
- Aributo
- Relaciones

Modelo lógico

Es un modelo sin base de datos que describe aspectos de las necesidades de recopilación de datos de una organización y las relaciones entre esos aspectos (IBM Documentation, 2021).

- Registros – Objetos

Modelo físico

Es un modelo específico de base de datos que figuriza objetos de datos relacionales (como tablas, columnas, cifras principales y claves externas) y sus correspondencias (IBM Documentation, 2021).

- Formato de las relaciones
- Distribución de los ficheros
- Porcedimiento de acceso

Arquitectura del software

Es un patrón para cualquier solución general y aprovechable a un problema de ingeniería de software en un contexto dado, son similares a los patrones usados en programación, pero especializados para estructuras de nivel superior y más general (Arquitectura de Software: Qué Es Y Qué Tipos Existen, 2022).

Importancia de la arquitectura de software

Al elegir un modelo arquitectónico, siempre hay varias cuestiones que determinan el uso final del software:

- Coste

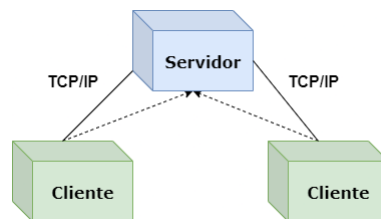
- Tiempo de desarrollo
- Número de usuarios
- Nivel de aislamiento

Patrones de arquitectura de software

Patrón cliente-servidor. El modelo cliente-servidor es muy utilizado, especialmente en el desarrollo de sitios web y servicios en línea, y se basa en la idea de que hay un servidor (que brinda un servicio) y varios clientes que envían solicitudes al servidor y reciben respuestas. Idéntico (Arquitectura de Software: Qué Es Y Qué Tipos Existen, 2022).

Figura 2

Patrón cliente-servidor.



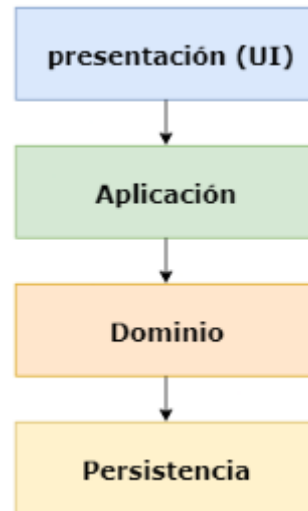
Nota. Debemos tener en cuenta los patrones de arquitectura de software. Tomado de (Arquitectura de Software: Qué Es Y Qué Tipos Existen, 2022).

Patrón de capas. En este esquema, la estructura del programa se divide en capas que representan una subtarea, cada capa perteneciente a un nivel diferente de abstracción. Cada capa está diseñada para proporcionar servicios a la siguiente capa superior (Arquitectura de Software: Qué Es Y Qué Tipos Existen, 2022).

Las siguientes capas son comúnmente utilizadas:

Figura 3

Representación de las capas.



Nota. Debemos tener en cuenta las capas para la arquitectura del software. Tomado de (Arquitectura de Software: Qué Es Y Qué Tipos Existen, 2022).

Patrón master-slave. Este modo consta de dos grupos, el primer grupo se denomina maestro y el segundo grupo se denomina grupo esclavo. Los esclavos ejecutan tareas iniciadas por el maestro, calculan los resultados y los devuelven al maestro, quien los inicia, almacena o procesa. Esto se hace para que un lado autorice y controle los cálculos necesarios, mientras que el otro maneja estas decisiones de manera agnóstica (Arquitectura de Software: Qué Es Y Qué Tipos Existen, 2022).

Patrón modelo-vista-controlador (MVC). Este conocido patrón, también conocido como patrón MVC, divide una aplicación interactiva en tres partes separadas:

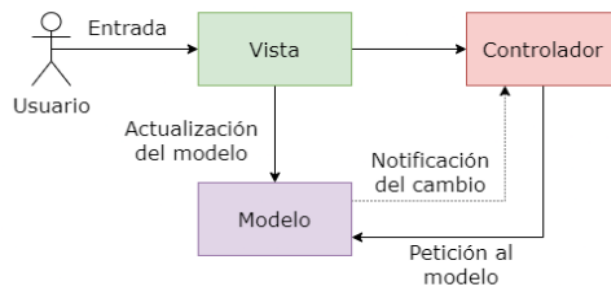
- **Modelo:** Contiene funcionalidad y datos básicos.

- **Vistas:** Para mostrar información al usuario, siempre es posible definir una o más vistas para la misma aplicación.
- **Controlador:** maneja la entrada del usuario. Esto se hace para separar la representación interna de la información de la forma en que el usuario proporciona y recibe información. Separa los componentes y permite la reutilización eficiente del código.

Este patrón es muy popular en el desarrollo de aplicaciones web, ya sea back-end o front-end, y es el patrón básico en muchos frameworks famosos (como Angular) y algunos lenguajes como Java y Same as Spring (Arquitectura de Software: Qué Es Y Qué Tipos Existen, 2022).

Figura 4

Modelo vista controlador (MVC)

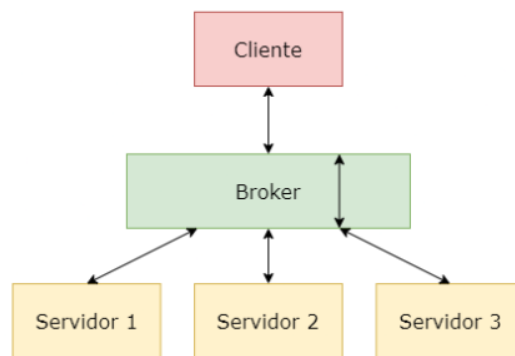


Nota. Debemos considerar el MVC para la arquitectura de software. Tomado de (Arquitectura de Software: Qué Es Y Qué Tipos Existen, 2022).

Patrón broker. Este modelo se utiliza para crear sistemas distribuidos con componentes desacoplados. Estos componentes pueden interactuar entre sí invocando servicios remotos de tal manera que anuncian sus capacidades, solicitan servicios y un componente llamado servidor proxy es responsable de coordinar la comunicación entre los componentes. Entonces puede ver su similitud con el patrón maestro-servicio (Arquitectura de Software: Qué Es Y Qué Tipos Existen, 2022).

Figura 5

Patrón broker.



Nota. Debemos considerar el patrón broker para la arquitectura de software. Tomado de (Arquitectura de Software: Qué Es Y Qué Tipos Existen, 2022).

FrontEnd

El front-end es la parte del programa o dispositivo a la que el usuario puede acceder directamente. Se trata tanto de tecnologías de diseño como de desarrollo web que se ejecutan en el navegador y se encargan de interactuar con los usuarios. Los desarrolladores front-end son los que crean los componentes visuales de un sitio web. Formatean imágenes, animaciones e interacciones en sitios web (Qué Es Frontend Y Backend: Diferencias Y Características - Platzi, 2023).

BackEnd

Un backend es una capa de software de acceso a datos o cualquier dispositivo al que el usuario no puede acceder directamente. Además, contiene la lógica de la aplicación para procesar dichos datos. El backend también accede al servidor, una aplicación especializada que comprende cómo el navegador realiza solicitudes.

Los desarrolladores de back-end son responsables del correcto funcionamiento de la lógica del sitio web, la transferencia segura de información y que el rendimiento de las aplicaciones no afecte la experiencia del usuario (Qué Es Frontend Y Backend: Diferencias Y Características - Platzi, 2023).

Diferencias entre FrontEnd y Backend

El desarrollo front-end es el proceso de hacer que un sitio web funcione y se vea bien en un navegador. Esto generalmente lo hacen los diseñadores y desarrolladores y, a veces, los especialistas en marketing. Los desarrolladores front-end pueden codificar usando HTML, CSS y JavaScript para integrar diseños del lado del cliente con aplicaciones, o pueden usar marcos como AngularJS (Qué Es Frontend Y Backend: Diferencias Y Características - Platzi, 2023).

El desarrollo de back-end es el proceso de administrar el almacenamiento de datos y el acceso a los datos de las bases de datos para mostrarlos en una página web que los usuarios pueden usar en cualquier dispositivo. Los desarrolladores de back-end a menudo necesitan dominar habilidades como Java, C, Ruby on Rails, PHP o Python, mientras que los desarrolladores de front-end pueden necesitar conocer algunos de estos lenguajes. Sin embargo, a menudo usan herramientas o marcos como Bootstrap o Tailwind que les permiten crear rápida y fácilmente sus propias interfaces web (Qué Es Frontend Y Backend: Diferencias Y Características - Platzi, 2023).

Capítulo III

Desarrollo del tema

Metodología SCRUM

Inicialmente, se planeó distribuir actividades a los miembros. Organizar las fases realizadas en el proceso global de implementación de una aplicación. Se llevarán a cabo reuniones diarias para revisar el progreso. Todas las actividades tienen tiempos estimados, incluidas las fechas de inicio y finalización, que deben cumplirse estrictamente.

Tabla 2*Metodología SCRUM*

FASE	PROCESO	RESPONSABLE	FECHA INICIO	FECHA FIN	DURACIÓN	STATUS	PROCESO
PLANIFICACIÓN	Identificar a los beneficiarios	Nataly	24/10/2022	28/10/2022	5 días	100%	100%
	Crear un plan de proyecto	Nataly	31/10/2022	04/10/2022	5 días	100%	100%
	Planificar la Gestión del Alcance	Jonathan	07/11/2022	11/11/2022	5 días	100%	100%
	Estimar la duración de las actividades	Jonathan	14/11/2022	18/11/2022	5 días	100%	100%
	Planificación de Gestión de cronograma	Nataly	21/11/2022	25/11/2022	5 días	100%	100%
	Planificación de Creación de cronograma	Jonathan	28/11/2022	02/12/2022	5 días	100%	100%
	Planificación de Desarrollo de Cronograma	Jonathan	05/12/2022	09/12/2022	5 días	100%	100%
EJECUCIÓN	Planificar gestión	Nataly	12/12/2022	16/12/2022	5 días	100%	100%
	Dirigir proyecto	Nataly	19/12/2022	06/01/2023	17 días	100%	100%

	PROCESO	RESPONSABLE	FECHA INICIO	FECHA FIN	DURACIÓN	STATUS	PROCESO
	Gestionar proyecto	Jonathan	09/01/2023	13/01/2023	5 días	100%	100%
	Control de alcance	Jonathan	16/01/2023	20/01/2023	5 días	100%	100%
MONITOREO	Monitorear	Nataly	23/01/2023	24/01/2023	2 días	100%	100%
	Control del cronograma	Nataly	25/01/2023	26/01/2023	1 día	100%	100%
CIERRE	Fin del proyecto	Todos	27/01/2023	27/01/2023	1 día	100%	100%

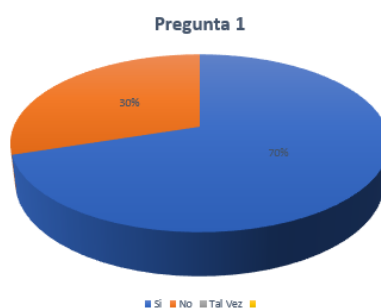
Análisis de Encuesta

Encuesta Administrador y Personal de la Radio Latacunga FM-AM

Pregunta 1 ¿Tiene conocimiento acerca de las aplicaciones web?

Figura 6

Encuesta pregunta 1

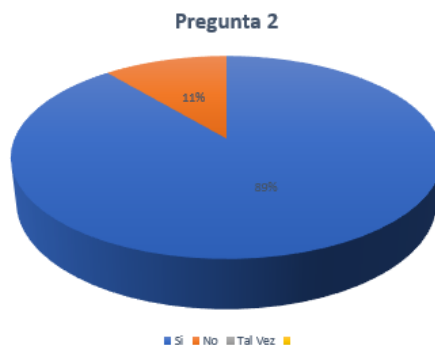


Nota. Mediante la encuesta realizada se obtuvo un 70% de la población tiene conocimiento sobre que es una aplicación web.

Pregunta 2 ¿Estima usted que se debería gestionar y controlar los mantenimientos en los equipos informáticos de la Radio?

Figura 7

Encuesta pregunta 2

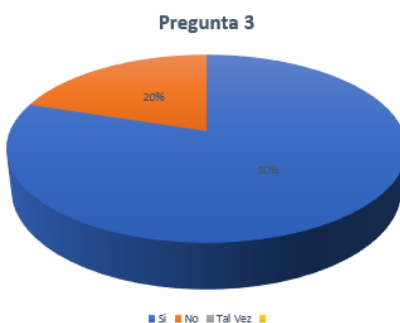


Nota. Mediante la encuesta realizada se obtuvo un 89% de la población estima la gestión y control de los mantenimientos.

Pregunta 3 ¿Considera necesario la implementación de una aplicación para tener un plan de mantenimiento de los equipos informáticos?

Figura 8

Encuesta pregunta 3

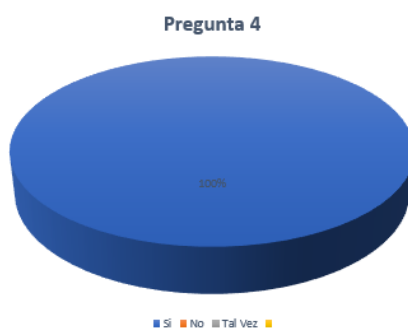


Nota. Mediante la encuesta realizada se obtuvo un 80% de la población que considera necesario la implementación de una aplicación web para gestionar los mantenimientos.

Pregunta 4 ¿Cree usted que los técnicos encargados de los mantenimientos utilizarían la aplicación web de la Radio Latacunga?

Figura 9

Encuesta pregunta 4

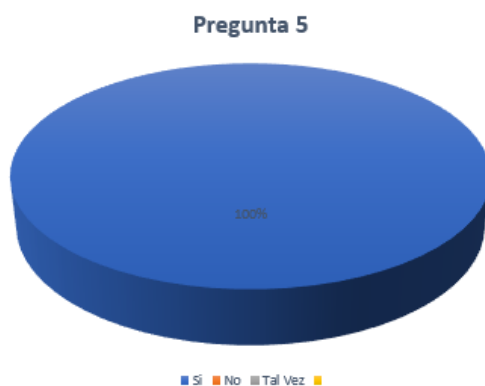


Nota. Mediante la encuesta realizada se obtuvo un 100% de la población que creen que los técnicos si utilizarían la aplicación web.

Pregunta 5 ¿Estima que el aplicativo web facilitaría el gestionamiento de los mantenimientos?

Figura 10

Encuesta pregunta 5

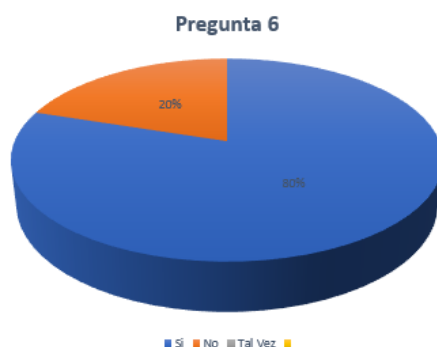


Nota. Mediante la encuesta realizada se obtuvo un 100% de la población que creen que el aplicativo web si facilitaría el gestionamiento de los mantenimientos.

Pregunta 6 ¿Considera usted que la aplicación web favorece al encargado de la Radio para prolongar la vida útil de los equipos informáticos?

Figura 11

Encuesta pregunta 6

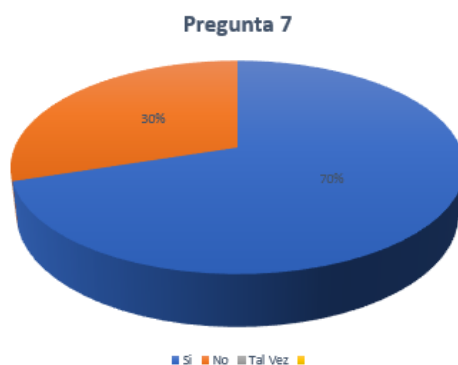


Nota. Mediante la encuesta realizada se obtuvo un 80% de la población que creen que el aplicativo web si prologaría la vida útil de los equipos informáticos.

Pregunta 7 ¿Considera usted que esta implementación del aplicativo web es una herramienta eficiente?

Figura 12

Encuesta pregunta 7

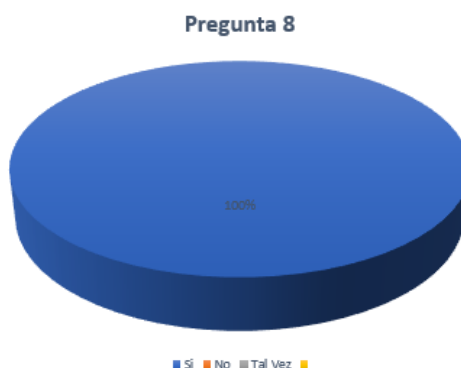


Nota. Mediante la encuesta realizada se obtuvo un 70% de la población que creen que la implementación del aplicativo web es una herramienta eficiente.

Pregunta 8 ¿Cree que con esta implementación se mejorara el funcionamiento de cada equipo informático?

Figura 13

Encuesta pregunta 8

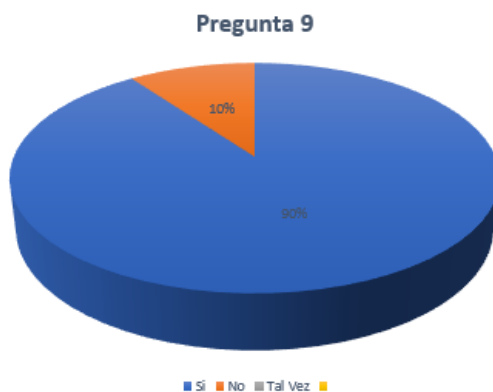


Nota. Mediante la encuesta realizada se obtuvo un 100% de la población que creen que con la implementación del aplicativo web se mejorara el funcionamiento de los equipos informáticos.

Pregunta 9 ¿Cree usted que la Radio Latacunga debe implementar esta tecnología para gestionar el mantenimiento de equipos informáticos?

Figura 14

Encuesta pregunta 9



Nota. Mediante la encuesta realizada se obtuvo un 90% de la población que creen se debe implementar el aplicativo web en la Radio Latacunga.

Requerimiento

Requerimiento de negocio

Inicialmente, para identificar las necesidades de la Radio Latacunga FM-AM, a la reunión asistieron los encargados de la Radio Latacunga FM-AM, en este caso el Padre Manolo Prado y el Ing. Patricio Aldana, quienes contaron el deseo de la introducción de esta nueva tecnología en la Radio Latacunga FM-AM, con antecedente de que uno de los encargados de la Radio Latacunga lo realizaba anualmente, sin mostrar los datos reales, además nos comentaron sobre la historia de cómo comenzó la Radio Latacunga FM-AM y la importancia de esta nueva tecnología en la radio. En la reunión, discutimos a detalle los requisitos generales acerca de la aplicación, así también de las diferentes limitaciones que tiene la Radio Latacunga FM-AM,

también hablamos acerca de las ideas que tenemos y cómo pensamos realizar la implementación de la aplicación en la radio.

Análisis del mercado

Se realizó un estudio comparando las aplicaciones existentes con las funciones solicitadas, lo cual es necesario tener conocimiento de cómo se establece el mercado de aplicaciones en base a su funcionamiento, con esto hemos podido asegurar los servicios necesarios para dar solución a lo que el cliente ha solicitado, así como el estudio de diferentes herramientas digitales las cuales brindan el soporte necesario, entre las aplicaciones de mercado se pudieron encontrar: SafetyCulture y Maximo Asset Management.

Después de revisar el mercado, se recomienda desarrollar una arquitectura de software para ayudar a satisfacer las necesidades de los clientes, por tal motivo nos centraremos en una arquitectura de microservicios.

Funcionalidades

Las principales funcionalidades que va a tener el aplicativo web son las siguientes:

- Deben existir dos roles de administrador y usuario.

Rol De Usuario

- Va a poder realizar una solicitud de mantenimiento de equipo y cualquiera de sus componentes, es decir, monitor, teclado, mouse o periféricos como parlantes, audífonos, micrófonos y consolas.
- Se debe tener en cuenta q las solicitudes pueden ser normales o urgentes, normales en el sentido q no necesita una atención inmediata y urgente en el caso de q sea imposible trabajar.

Rol De Administrador

- El administrador va a poder crear cualquier perfil de usuario.
- El administrador va a poder ver cualquier perfil de usuario.
- El administrador va a poder editar cualquier perfil de usuario.
- El administrador va a poder eliminar cualquier perfil de usuario.
- El administrador va a poder visualizar inmediatamente cualquier solicitud urgente que exista a través de una alerta.
- El administrador va a poder ver las solicitudes que se encuentren pendientes.
- El administrador va a poder aprobar cualquier solicitud que esté pendiente.
- El administrador va a poder eliminar cualquier solicitud que esté pendiente.
- El administrador va a poder crear una orden de mantenimiento de las solicitudes aprobadas.
- El administrador va a poder imprimir la orden de mantenimiento generada en solicitudes aprobadas.
- El administrador va a poder eliminar las solicitudes aprobadas.
- El administrador va a poder finalizar la solicitud que se encuentren ya generadas su orden de mantenimiento.
- El administrador va a poder ver las solicitudes finalizadas.
- El administrador va a poder imprimir las solicitudes finalizadas.
- El administrador va a poder eliminar las solicitudes finalizadas.
- El administrador puede realizar filtros por fecha a las solicitudes de mantenimiento.
- El administrador puede realizar filtros por área a las solicitudes de mantenimiento.

- El administrador va a poder añadir equipos nuevos.
- El administrador va a poder ver los detalles de los equipos en funcionamiento.
- El administrador va a poder actualizar cualquier detalle de los equipos.
- El administrador va a poder dar de baja a cualquier componente del equipo.
- El administrador va a poder eliminar los equipos.
- El administrador puede filtrar por nombre los equipos.
- El administrador puede filtrar por área los equipos.
- El administrador va a poder crear un plan de mantenimiento preventivo.
- El administrador va a poder editar un plan de mantenimiento preventivo.
- El administrador va a poder eliminar un plan de mantenimiento preventivo.
- El administrador puede filtrar por partes los planes de mantenimiento preventivo.
- El administrador puede filtrar por nombre los planes de mantenimiento preventivo.
- El administrador va a poder crear los planes de mantenimiento correctivos y adaptativos.
- El administrador va a poder ver los planes de mantenimiento correctivos y adaptativos.
- El administrador va a poder eliminar los planes de mantenimiento correctivos y adaptativos.
- El administrador puede realizar filtros por nombre a los planes de mantenimiento no rutinario.
- El administrador puede realizar filtros por parte a los planes de mantenimiento no rutinario.

- El administrador va a poder visualizar los futuros mantenimientos preventivos en un calendario.

Entorno Operativo

El aplicativo web va a ser diseñado en dos instancias primero se va a generar una API REST haciendo uso de NODE JS como backend y se va a consumir la API desde el frontend en donde se va a hacer uso de VUE JS como base de datos se va a hacer uso de MySQL para el desarrollo local al momento de construir el aplicativo es necesario q sea subido al servidor de SINET.

Requisitos Funcionales

Funcionalidad 1. Autenticación de usuarios

Descripción: El sistema tendrá los roles que serán de Usuario y Administrador.

Prioridad: Alta

REQ-1: Deben Existir Dos Roles de Administrador y Usuario.

Un usuario administrador se va a encontrar creado y a partir de este usuario administrador es donde se va a seguir creando los otros usuarios ya sea con rol de administrador o rol de usuario. Sólo se aceptarán estos dos tipos de Roles.

Funcionalidad 2. Generar solicitudes desde Usuario

Descripción: Las personas que tengan el rol de usuario van a ser capaces de realizar solicitudes para el mantenimiento.

Prioridad: Alta

REQ-1: Solicitud de Mantenimiento. Va a poder realizar una solicitud de mantenimiento de equipo y cualquiera de sus componentes, es decir, monitor, teclado, mouse o periféricos como parlantes, audífonos, micrófonos y consolas.

REQ-2: Tipos de Solicitudes. Se debe tener en cuenta q las solicitudes pueden ser normales o urgentes, normales en el sentido q no necesita una atención inmediata y urgente en el caso de q sea imposible trabajar.

Funcionalidad 3.

El sistema permitirá la gestión de usuarios desde el administrador

Descripción: El administrador del aplicativo podrá tener la capacidad de realizar un CRUD, es decir, crear, ver, editar y eliminar cualquier usuario.

Prioridad: Alta

REQ-1: Creación de Usuarios

Dentro de las opciones que tiene el perfil de administrador, podrá crear perfiles de usuarios que tengan el nombre de usuario, nombre, apellido, email, área, rol y su contraseña.

REQ-2: Visualización de Usuarios

Dentro de las opciones que tiene el perfil de administrador, podrá ver los perfiles de usuarios con su nombre de usuario, nombre y apellido, email y su área.

REQ-3: Edición de Usuarios

Dentro de las opciones que tiene el perfil de administrador, podrá editar perfiles de usuarios en donde lo único que puede editar será el rol, área y la contraseña.

REQ-4: Eliminación de Usuarios

Dentro de las opciones que tiene el perfil de administrador, podrá eliminar perfiles de usuarios que se encuentre obsoletos siempre con una notificación de si está seguro eliminar.

Funcionalidad 4. El sistema permitirá la gestión de solicitudes de mantenimiento de los usuarios

Descripción: Las solicitudes que realicen los usuarios para cualquier mantenimiento se podrán administrar desde el perfil de administrador.

Prioridad: Alta

REQ-1: Visualización de Solicitudes Urgentes

El administrador va a poder visualizar inmediatamente cualquier solicitud urgente que exista a través de una alerta, además de tener en la página principal cualquier solicitud urgente que esté al momento.

REQ-2: Visualización de Solicitudes Pendientes

Dentro del software en el apartado de solicitudes se va a tener una tablita que permita la navegación entre solicitudes pendientes aprobadas y finalizadas, para lo cual, en el apartado de pendientes se podrá ver las solicitudes que se encuentre pendientes esperando la aprobación.

REQ-3: Aprobación de Solicitudes Pendientes

Dentro del software en el apartado de solicitudes se va a tener una tablita que permita la navegación entre solicitudes pendientes, aprobadas y finalizadas, para lo cual, en el apartado de pendientes se tendrá una opción para poder aprobar la solicitud de mantenimiento.

REQ-4: Eliminación de Solicitudes Pendientes

Dentro del software en el apartado de solicitudes se va a tener una tablita que permita la navegación entre solicitudes pendientes, aprobadas y finalizadas, para lo cual, en el apartado de pendientes se tendrá una opción para poder eliminar la solicitud de mantenimiento.

REQ-5: Creación de Orden de Mantenimiento

Dentro del software en el apartado de solicitudes se va a tener una tablita que permita la navegación entre solicitudes pendientes, aprobadas y finalizadas, para lo cual, en el apartado de aprobadas se tendrá una opción para poder generar la orden de mantenimiento respectiva.

REQ-6: Impresión de Orden de Mantenimiento Generada

Dentro del software en el apartado de solicitudes se va a tener una tablita que permita la navegación entre solicitudes pendientes, aprobadas y finalizadas, para lo cual, en el apartado de aprobadas una vez generada la orden se tendrá una opción para imprimir dicha orden.

REQ-7: Eliminación de Solicitudes Aprobadas

Dentro del software en el apartado de solicitudes se va a tener una tablita que permita la navegación entre solicitudes pendientes, aprobadas y finalizadas, para lo cual, en el apartado de aprobadas se tendrá una opción para poder eliminar cualquier solicitud q esté aprobada.

REQ-8: Finalización de Solicitudes Aprobadas

Dentro del software en el apartado de solicitudes se va a tener una tablita que permita la navegación entre solicitudes pendientes, aprobadas y finalizadas, para lo cual, en el apartado de aprobadas una vez que el mantenimiento se haya realizado se ingresarán los datos necesarios para finalizar la solicitud.

REQ-9: Visualización de Solicitudes Finalizadas

Dentro del software en el apartado de solicitudes se va a tener una tablita que permita la navegación entre solicitudes pendientes, aprobadas y finalizadas, para lo cual, en el apartado de finalizadas se podrá observar todas las solicitudes que ya se encuentren finalizadas.

REQ-10: Impresión de Solicitudes Finalizadas

Dentro del software en el apartado de solicitudes se va a tener una tablita que permita la navegación entre solicitudes pendientes, aprobadas y finalizadas, para lo cual, en el apartado de finalizadas se tendrá una opción para poder generar un pdf e imprimir cualquier solicitud que se encuentre finalizada.

REQ-11: Eliminación de Solicitudes Finalizadas

Dentro del software en el apartado de solicitudes se va a tener una tablita que permita la navegación entre solicitudes pendientes, aprobadas y finalizadas, para lo cual, en el apartado de finalizadas se tendrá una opción que permita eliminar cualquier solicitud ya finalizada siempre teniendo en cuenta su cuadro de confirmación para realizar esta acción.

REQ-12: Filtros por Fecha en Solicitudes

Dentro del software en el apartado de solicitudes se va a tener una tablita que permita la navegación entre solicitudes pendientes, aprobadas y finalizadas, para lo cual, en cada apartado de visualización de solicitudes se tendrá un filtro por fecha para poder ver las más recientes o las más viejas.

REQ-13: Filtros por Área en Solicitudes

Dentro del software en el apartado de solicitudes se va a tener una tablita que permita la navegación entre solicitudes pendientes, aprobadas y finalizadas, para lo cual, en cada apartado de visualización de solicitudes se tendrá un filtro por área para poder ver organizadamente las solicitudes hechas por área.

Funcionalidad 5. El sistema permitirá la gestión de mantenimiento de equipos

Descripción: El sistema tendrá el apartado de mantenimiento en donde podrá realizar acciones con los equipos, realizar planes de mantenimiento para el mantenimiento preventivo,

trabajos no rutinarios para el mantenimiento adaptativo y correctivo y por último tener un acceso a un calendario que indicará los días de mantenimiento.

Prioridad: Alta

REQ-1: Adición de Equipos Nuevos

Dentro del software en el apartado de mantenimiento se va a tener un menú de navegación que permita el acceso a las distintas opciones como equipos, planes de mantenimiento, trabajos no rutinarios y calendario, para lo cual, en el apartado de equipos se va a poder agregar cualquier equipo, el equipo se divide en pc, monitor, teclado, mouse, micrófono, parlantes, consola y otros entre estos otros pueden ser ya sea servidores o impresoras.

REQ-2: Visualización de Detalles de Equipos

Dentro del software en el apartado de mantenimiento se va a tener un menú de navegación que permita el acceso a las distintas opciones como equipos, planes de mantenimiento, trabajos no rutinarios y calendario, para lo cual, en el apartado de equipos se va a tener una tabla en donde se va a mostrar el área, nombre y el estado de si está activo o inactivo.

REQ-3: Actualización de Detalles del Equipo

Dentro del software en el apartado de mantenimiento se va a tener un menú de navegación que permita el acceso a las distintas opciones como equipos, planes de mantenimiento, trabajos no rutinarios y calendario, para lo cual, en el apartado de equipos se va a poder editar cualquier componente desde el pc hasta añadir otros.

REQ-4: Eliminación de Componentes del Equipo

Dentro del software en el apartado de mantenimiento se va a tener un menú de navegación que permita el acceso a las distintas opciones como equipos, planes de mantenimiento, trabajos no rutinarios y calendario, para lo cual, en el apartado de equipos y

dentro de detalles del equipo se va a tenerla opción de eliminar cualquiera de los componentes que posee.

REQ-5: Eliminación de Equipos

Dentro del software en el apartado de mantenimiento se va a tener un menú de navegación que permita el acceso a las distintas opciones como equipos, planes de mantenimiento, trabajos no rutinarios y calendario, para lo cual, en el apartado de equipos se tendrá la opción de eliminar todo el equipo.

REQ-6: Filtro por Área en Equipos

Dentro del software en el apartado de mantenimiento se va a tener un menú de navegación que permita el acceso a las distintas opciones como equipos, planes de mantenimiento, trabajos no rutinarios y calendario, para lo cual, en el apartado de equipos se tendrá un filtro para ver los equipos que maneja cada área.

REQ-7: Filtro por Nombre en Equipos

Dentro del software en el apartado de mantenimiento se va a tener un menú de navegación que permita el acceso a las distintas opciones como equipos, planes de mantenimiento, trabajos no rutinarios y calendario, para lo cual, en el apartado de equipos se tendrá un filtro por nombre para encontrar más rápido el equipo.

REQ-8: Creación de Plan de Mantenimiento Preventivo

Dentro del software en el apartado de mantenimiento se va a tener un menú de navegación que permita el acceso a las distintas opciones como equipos, planes de mantenimiento, trabajos no rutinarios y calendario, para lo cual, en el apartado de planes de mantenimiento se va a poder crear un plan de mantenimiento dándole un nombre, asignándole una actividad, seleccionando las partes que van a tener acceso a este plan, la frecuencia,

duración, prioridad y responsable de la actividad así como su procedimiento y si necesita días de paro.

REQ-9: Edición de Planes de Mantenimiento Preventivo

Dentro del software en el apartado de mantenimiento se va a tener un menú de navegación que permita el acceso a las distintas opciones como equipos, planes de mantenimiento, trabajos no rutinarios y calendario, para lo cual, en el apartado de planes de mantenimiento se va a poder editar los planes que ya se encuentre creados, lo único que no se podrá editar es el nombre y la actividad que realiza de ahí tanto las partes como la duración, frecuencia, prioridad, responsable, días de paro y procedimiento que se realiza podrá ser editable.

REQ-10: Eliminación de Planes de Mantenimiento Preventivo

Dentro del software en el apartado de mantenimiento se va a tener un menú de navegación que permita el acceso a las distintas opciones como equipos, planes de mantenimiento, trabajos no rutinarios y calendario, para lo cual, en el apartado de planes de mantenimiento se tendrá la opción para poder eliminar por completo cualquier plan que esté creado.

REQ-11: Filtro por Nombre en Planes de Mantenimiento Preventivo

Dentro del software en el apartado de mantenimiento se va a tener un menú de navegación que permita el acceso a las distintas opciones como equipos, planes de mantenimiento, trabajos no rutinarios y calendario, para lo cual, en el apartado de planes de mantenimiento se tendrá un filtro por nombre para encontrar más rápido el plan de mantenimiento que se necesita.

REQ-12: Filtro por Partes en Planes de Mantenimiento Preventivo

Dentro del software en el apartado de mantenimiento se va a tener un menú de navegación que permita el acceso a las distintas opciones como equipos, planes de mantenimiento, trabajos no rutinarios y calendario, para lo cual, en el apartado de planes de mantenimiento se tendrá un filtro por partes para encontrar el mejor plan que necesitemos para un componente específico.

REQ-13: Visualización de Planes de Mantenimiento Preventivo

Dentro del software en el apartado de mantenimiento se va a tener un menú de navegación que permita el acceso a las distintas opciones como equipos, planes de mantenimiento, trabajos no rutinarios y calendario, para lo cual, en el apartado de planes de mantenimiento se podrá ver los planes ya creados, así como su duración, nombre, partes, actividades, frecuencia y el estado del plan.

REQ-14: Creación de Planes de Mantenimiento Correctivos y Adaptativos

Dentro del software en el apartado de mantenimiento se va a tener un menú de navegación que permita el acceso a las distintas opciones como equipos, planes de mantenimiento, trabajos no rutinarios y calendario, para lo cual, en el apartado de trabajos no rutinarios se va a tener una opción que permita realizar esto seleccionando el tipo de mantenimiento, dándole un nombre, asignándole una actividad, seleccionando la parte o partes a corregir o cambiar, la fecha de cuándo se realizará, duración, prioridad y responsable de la actividad así como su procedimiento y si necesita días de paro.

REQ-15: Visualización de Planes de Mantenimiento Correctivos y Adaptativos

Dentro del software en el apartado de mantenimiento se va a tener un menú de navegación que permita el acceso a las distintas opciones como equipos, planes de mantenimiento, trabajos no rutinarios y calendario, para lo cual, en el apartado de trabajos no

rutinarios se tendrá una tabla que nos mostrará la información del plan de mantenimiento, la fecha que se realizará la parte o partes, nombre, actividades y duración de la actividad, así como su tipo si es correctivo o adaptativo.

REQ-16: Eliminación de Planes de Mantenimiento Correctivos y Adaptativos

Dentro del software en el apartado de mantenimiento se va a tener un menú de navegación que permita el acceso a las distintas opciones como equipos, planes de mantenimiento, trabajos no rutinarios y calendario, para lo cual, en el apartado de trabajos no rutinarios se tendrá la opción para eliminar cualquier trabajo no rutinario ya creado.

REQ-17: Edición de Planes de Mantenimiento Correctivos y Adaptativos

Dentro del software en el apartado de mantenimiento se va a tener un menú de navegación que permita el acceso a las distintas opciones como equipos, planes de mantenimiento, trabajos no rutinarios y calendario, para lo cual, en el apartado de trabajos no rutinarios se tendrá la opción de editar cualquiera de estos trabajos no rutinarios correctivos o adaptativos en la ventana de edición lo único que no puede cambiar es su nombre, actividad y tipo.

REQ-18: Filtro por Nombre en Trabajos no Rutinarios

Dentro del software en el apartado de mantenimiento se va a tener un menú de navegación que permita el acceso a las distintas opciones como equipos, planes de mantenimiento, trabajos no rutinarios y calendario, para lo cual, en el apartado de trabajos no rutinarios se tendrá un filtro por nombre para encontrar de manera más rápida el plan de mantenimiento que necesitamos.

REQ-19: Filtro por Componente en Trabajos no Rutinarios

Dentro del software en el apartado de mantenimiento se va a tener un menú de navegación que permita el acceso a las distintas opciones como equipos, planes de

mantenimiento, trabajos no rutinarios y calendario, para lo cual, en el apartado de trabajos no rutinarios se tendrá un filtro por componente para encontrar de manera más cómoda y rápida el plan de mantenimiento adaptativo o correctivo.

REQ-20: Visualización de Futuros Mantenimientos Preventivos En Calendario

Dentro del software en el apartado de mantenimiento se va a tener un menú de navegación que permita el acceso a las distintas opciones como equipos, planes de mantenimiento, trabajos no rutinarios y calendario, para lo cual, en el apartado de calendario se podrá ver de color verde todos los mantenimientos preventivos próximos en el calendario.

REQ-21: Visualización de Futuros Mantenimientos Correctivos y Adaptativos en Calendario

Dentro del software en el apartado de mantenimiento se va a tener un menú de navegación que permita el acceso a las distintas opciones como equipos, planes de mantenimiento, trabajos no rutinarios y calendario, para lo cual, en el apartado de calendario se podrá ver de color rojo los mantenimientos adaptativos y de color azul los mantenimientos correctivos próximos en el calendario.

Requisitos No Funcionales

Eficiencia

- El aplicativo debe ser capaz de alertar N solicitudes urgentes.
- Toda funcionalidad del aplicativo y transacción de negocio debe responder al usuario en menos de 5 segundos.
- El aplicativo debe ser capaz de operar adecuadamente con hasta 50 usuarios con sesiones concurrentes.
- Los datos modificados en la base de datos deben ser actualizados para todos los usuarios que acceden en menos de 5 segundos.

Seguridad lógica y de datos

- Los permisos de acceso al aplicativo podrán ser cambiados solamente por el administrador de acceso a datos.

Usabilidad

- La tasa de errores cometidos por el usuario deberá ser menor del 1% de las solicitudes totales ejecutadas en el aplicativo.
- El aplicativo debe proporcionar mensajes de error que sean informativos y orientados a usuario final.
- La aplicación web debe poseer un diseño “Responsive” a fin de tener una adecuada visualización en múltiples computadores personales, dispositivos tableta y teléfonos inteligentes a pesar de no estar enfocado a dispositivos móviles.
- El aplicativo debe poseer interfaces gráficas bien formadas.

Análisis

Análisis del proceso de negocios

En este proceso se ejecutará un análisis de los procesos que se van a ejecutar dentro de la Radio Latacunga FM-AM por lo que se llevó a cabo una recopilación de los mismos entre los cuales se encuentran:

Se necesita poder realizar un mejor manejo de los mantenimientos preventivos, correctivos y adaptativos dentro de la Radio Latacunga ya que no cuenta con un registro de mantenimientos para poder llevar un seguimiento de estos datos, por lo que se requiere contar con un sistema que nos permita recopilar y proteger esta información para que solo el personal encargado pueda ejecutar cambios en la información de los mantenimientos

El segundo proceso es la asignación de nuevos mantenimientos, en este proceso se deberá generar una nueva fecha para el siguiente mantenimiento para así evitar fallos en los equipos tecnológicos de la radio.

Desarrollar casos de uso

Tabla 3

Caso de estudio Administrador

Caso de uso Administrador	
Servicios	Descripción
Crear usuarios	Puede crear nuevos usuarios para todos los trabajadores
Escritura de datos	Puede registrar datos
Actualización de datos	Puede Modificar datos
Lectura de datos	Poder visualizar todo el contenido de la aplicación

Nota. Este usuario es creado al iniciar el sistema.

Tabla 4

Caso de uso Usuario

Caso de uso Usuario	
Servicios	Descripción
Registro de mantenimientos (notificar mantenimiento al administrador)	Puede registrar la información del mantenimiento, pero siempre reportando al administrador
Revisar fechas de nuevos mantenimientos	Puede revisar fechas sobre nuevos mantenimientos
Actualización de datos (notificar al administrador)	Puede Modificar datos, pero notificando al administrador
Lectura de datos	Poder visualizar todo el contenido de los mantenimientos anteriores.

Nota. Es el usuario encargado de llevar a cabo los mantenimientos preventivos y correctivos de los equipos informáticos.

Investigación de la Oferta

Se procede a realizar una investigación en base a los datos obtenidos, la misma nos permitirá determinar cuáles son las tecnologías que se ajusta a las necesidades técnicas de la Radio Latacunga por lo que a continuación se enlistan algunas de las ofertas para los servicios que se requiere.

Servidores Web

En base a los datos obtenidos previamente, realizar una investigación para determinar qué tecnologías se pueden adaptar a las necesidades de la franquicia y las necesidades técnicas que se deben cubrir. Por ello, a continuación, se presentan las ofertas más relevantes para cada servicio deseado.

Tabla 5

Comparativa de ofertas de servidores web.

Oferta	Ventajas técnicas	Desventajas técnicas
Apache	<ul style="list-style-type: none"> • Software verídico y duradero. • Código abierto y arbitrario, incluso para el empleo comercial. 	<ul style="list-style-type: none"> • Problemas de acatamiento en alto tráfico. • Puede agendar inseguridades de estabilidad.
Django	<ul style="list-style-type: none"> • Ofrece un gran acatamiento y transigencia, pudiendo escalar planes de forma sencilla. • Adjunta una amplia multiplicidad de paquetes de librerías (más de 4000). • Ideal para aplicaciones en tiempo concreto. 	<ul style="list-style-type: none"> • Acreditación puede ser dudoso y dispersa. • Compleja para el progreso de una API REST o con sockets.
NodeJS	<ul style="list-style-type: none"> • Cuenta con un procurador de atadijos de más de 50.000 recursos. • Permite un alto nivel de adaptabilidad. 	<ul style="list-style-type: none"> • Deficiencia de librerías estándar, no cuenta con una base original. • Malgasta potencia de cálculo debido a sus casos asíncronos.

Bases de Datos. Para la base de datos se debe considerar aquellas que sean capaces de almacenar grandes cantidades información debido a la gran cantidad de equipos informáticos que existen en la Radio Latacunga FM-AM. Por lo que se decide considerar las ofertas de la tabla 6.

Tabla 6

Comparativa de ofertas en base de datos.

Oferta	Ventajas	Desventajas
Mongo DB	<ul style="list-style-type: none"> • Puede leer y escribir grandes cantidades de información porque es una base de datos no relacional. • Es ideal para trabajar con JavaScript • Garantiza la seguridad y permite trabajar con varias tablas de bases de datos. 	<ul style="list-style-type: none"> • No tiene Joins para consultas ya que puede tener problemas de relación entre tablas diferentes. • No es una base de datos adecuada para aplicaciones con transacciones complejas • Supera el consumo de memoria RAM para las instalaciones y utilización de software.
SQL Server	<ul style="list-style-type: none"> • Permite administrar los datos de manera fácil y eficiente para consultas. 	<ul style="list-style-type: none"> • Contiene diferentes bloqueos a nivel de página con un tamaño fijo y muy pequeño.
PostgreSQL	<ul style="list-style-type: none"> • Es un software estable y confiable. 	<ul style="list-style-type: none"> • No cuenta con un soporte oficial.

Frameworks o Librerías para Maquetado Frontend. Una aplicación web debe corresponder a la parte de la interfaz gráfica que permite al usuario interactuar con ella. Las ofertas más actuales del mercado se evalúan según el servidor web predefinido.

Tabla 7

Contraposición de ofertas en librerías para maquetado FrontEnd.

Oferta	Ventajas técnicas	Desventajas técnicas
ExpressJS	<ul style="list-style-type: none"> • Le da a su solicitud una ventaja, agrega análisis y encabezados automáticos. • Código de conversión para información de soporte. 	<ul style="list-style-type: none"> • Dado que el sistema es relativamente nuevo, ha tenido problemas durante su creciente desarrollo. • Documentación y una comunidad ligeramente ampliada.
NestJS	<ul style="list-style-type: none"> • Sus módulos y controladores facilitan la gestión de microservicios. • Debido a su estructura, es útil en el desarrollo de backend. 	<ul style="list-style-type: none"> • Su sintaxis puede volverse larga y confusa en el desarrollo front-end, lo que dificulta su implementación.
ReactJS	<ul style="list-style-type: none"> • La sintaxis es simple y fácil de leer si conoce HTML y Javascript. • Contiene un DOM virtual que puede estar compuesto en diferentes tipos de lenguajes de representación visual. 	<ul style="list-style-type: none"> • La falta de un estándar confunde el número de soluciones disponibles. • El aprendizaje efectivo de la manipulación del DOM lleva tiempo.

Servicios en la Nube. En el caso de una aplicación alojada, debe ser capaz de cumplir con los requisitos futuros del franquiciado. Para cumplir con esto, se determina el uso y las ofertas de los servicios en la nube.

Tabla 8

Contraposición de ofertas en servicios en la nube.

Oferta	Ventajas técnicas	Desventajas técnicas
Google Cloud	<ul style="list-style-type: none"> • Precio más competitivo en comparación con los competidores. • Se centra en la extensibilidad y la escalabilidad. 	<ul style="list-style-type: none"> • No tiene automatización CI/CD u otras características requeridas.
Amazon Web Services	<ul style="list-style-type: none"> • Implementar una manera fácil de usar CI/CD es muy útil para los equipos de DevOps. • Tiene amplias opciones de configuración, escalabilidad, instalación y más. 	<ul style="list-style-type: none"> • Su proceso de implementación es muy extenso. • No dispone de un gestor que facilite el despliegue de aplicaciones.

Evaluación de la oferta

A continuación, se muestran enteramente las tecnologías seleccionadas en función de los datos más relevantes compilados en el paso anterior.

NodeJS: Servidor Web. Su extensa biblioteca de herramientas en el administrador de paquetes NPM le permitirá implementar bibliotecas útiles en el proceso de diseño de aplicaciones web. Basado en Javascript, los datos JSON también se pueden manipular más fácilmente.

Su rendimiento es excelente y así lo demuestra la cantidad de desarrolladores que lo utilizan para desarrollar aplicaciones web. Esto reduce en gran medida las posibilidades de que el servidor web pueda cometer errores técnicos.

MongoDB Compass: Base de Datos. La base de datos necesitaba poder almacenar una gran cantidad de datos, y para este mismo propósito, se utilizó MongoDB como una base de datos no relacional que contenía información de documentos JSON. Es altamente compatible con los servidores NodeJS que se ejecutan a nivel de Javascript.

Para garantizar la disponibilidad de la base de datos, se utiliza MongoDB Atlas, un servicio de MongoDB que le permite alojar un clúster de MongoDB que cuenta con el soporte técnico de sus desarrolladores. Esto le permite mantener su información siempre disponible sin problemas de bloqueo.

Elasticsearch: Motor de Búsqueda y Base de Datos. Aunque puede considerarse una base de datos, no es recomendable para almacenar grandes cantidades de información. Pero es un complemento perfecto para MongoDB porque cuando MongoDB registra información, Elasticsearch indexará la información y proporcionará un sistema de consulta más rápido que MongoDB solo.

Además de agregar un poderoso motor de búsqueda que viene con él, que brinda asesoramiento instantáneo sobre productos a través de sinónimos, existen otras consideraciones para una búsqueda eficiente.

ReactJS: Librería para Maquetado Frontend. Una de las principales razones para usar ReactJS es que se puede enrutar a través de diferentes puntos finales sin perder los componentes renderizados previamente, lo que acelera el proceso de carga de la aplicación y lo hace sentir como una aplicación.

La fuerza de ReactJS radica en su uso del diseño modular, que te permite usar cada elemento que creas y adaptarlo a la versión móvil. Esto minimiza el código de diseño.

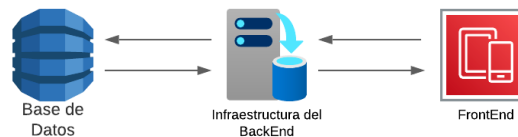
Arquitectura de servicios

Arquitectura Básica de la Aplicación. Define la arquitectura básica para la comunicación entre cada servicio central.

- Front-end: Responsable de presentar la interfaz gráfica que interactúa directamente con los usuarios.
- Backend: consta de microservicios que manejan toda la lógica de la aplicación. De esa forma, evitará la congestión del front-end que reduce el rendimiento.
- Base de datos: Almacena toda la información para su uso posterior.

Figura 15

Arquitectura primordial de una aplicación web.

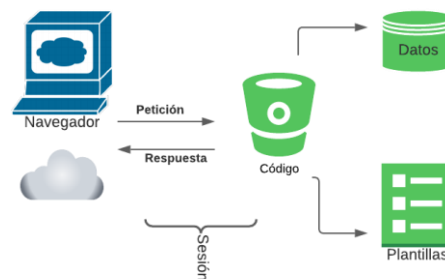


Arquitectura del Frontend. Como responsable de la interacción directa con el usuario, dispone de una interfaz gráfica para tal fin. Está integrado por el siguiente personal:

- NodeJS: servidor web.
- ReactJS: un lenguaje de maquetación.

Figura 16

Correspondencia de la aplicación web entre el backend y el usuario.



Arquitectura del Backend. Incluye proporcionar acceso a datos, servicios y otros sistemas que soportan el funcionamiento de la aplicación.

- Elastic: Pila compuesta por Elasticsearch.

Esto mantiene las aplicaciones escalables y siempre disponibles sin sacrificar el rendimiento.

Ingeniería de procesos de negocio

Para almacenar la información en la franquicia “Farmacias Marthita”, se debe contar con una arquitectura de software basada en microservicios que ayude al proceso, y el servicio de base de datos se encarga de recolectar la información utilizable. Necesitamos un servicio que nos ayude con la visualización de datos, por lo que primero usamos Elasticsearch, que se encarga de almacenar los datos del índice, para leer los datos de manera más eficiente y poder generar sugerencias de inteligencia de motor de búsqueda optimizadas.

Para los datos ya comentados, se representan gráficamente, así como representaciones estadísticas de la información responsable de la representación gráfica de los datos.

Documentación tradicional

Este manual es un documento técnico que introduce al usuario a la funcionalidad de la aplicación web de Radio Latacunga, donde se detalla interactivamente cada interfaz. El objetivo principal de esta guía es crear una guía clara para que los usuarios comiencen rápidamente sin leer mucho texto, siguiendo las imágenes detalladas paso a paso de la guía anterior.

Figura 17*Manual de Usuario*

Nota. Se menciona de la documentación de interfaces y su funcionamiento.

Diseñar casos de prueba

Tabla 9

Casos de prueba

ID Caso de prueba	Descripción de la prueba	Supuestos y condiciones previas	Datos de prueba	Pasos a ejecutar	Resultado Esperado
CP01	Verificar la aplicación para ver cómo funciona en Google Chrome	El navegador primero se instala en el dispositivo en el que necesitamos probar, el servicio se implementa en la nube para que se puedan realizar pruebas correctivas con una conexión a Internet.	Para realizar la prueba es necesario ingresar a este enlace http://localhost:5173/	Introducir la URL. Visualizar tu renderizado. Comprobar el contenido.	El tiempo de reacción debe ser inferior a 3 segundos. Renderiza los componentes construidos sin errores.
CP02	Verificar la aplicación para ver cómo funciona en Edge	El navegador primero se instala en el dispositivo en el que necesitamos probar, el servicio se implementa en la nube para que se puedan realizar pruebas correctivas con una conexión a Internet.	Para realizar la prueba es necesario ingresar a este enlace http://localhost:5173/	Introducir la URL. Visualizar tu renderizado. Comprobar el contenido.	El tiempo de reacción debe ser inferior a 3 segundos. Renderiza los componentes construidos sin errores.

ID Caso de prueba	Descripción de la prueba	Supuestos y condiciones previas	Datos de prueba	Pasos a ejecutar	Resultado Esperado
CP03	Verificar la aplicación para ver cómo funciona en Firefox	El navegador primero se instala en el dispositivo en el que necesitamos probar, el servicio se implementa en la nube para que se puedan realizar pruebas correctivas con una conexión a Internet.	Para realizar la prueba es necesario ingresar a este enlace http://localhost:5173/	Introducir la URL. Visualizar tu renderizado. Comprobar el contenido.	El tiempo de reacción debe ser inferior a 3 segundos. Renderiza los componentes construidos sin errores.
CP04	Confirmar cómo funcionan los tiempos de carga	Analiza la cantidad de usuarios simulados en segundo plano, en este caso define los tipos de pruebas de carga y rendimiento, aprende a medir el tiempo de carga utilizando la plataforma Google Page Speed Insights.	Para realizar la prueba es necesario ingresar a este enlace https://pagespeed.web.dev/	Introducir la URL. Visualizar tu renderizado. Comprobar el contenido.	El tiempo de reacción debe ser inferior a 3 segundos. Renderiza los componentes construidos sin errores.

Nota. Se reverbera casos en distintos campos para el correcto funcionamiento de la aplicación web.

Diseño de Interfaz de Usuario

El diseño gráfico define dos arquetipos básicos que le permiten diseñar su aplicación.

Diseño gráfico conceptual y experiencia de usuario. Se dibujó un croquis de la estructura básica y organización visual de cada elemento incluido en el proyecto.

. Para este boceto, utilizando la guía UX basada en la psicología humana, buscábamos una interfaz intuitiva que no causara dificultades a los usuarios objetivo en su tratamiento diario.

Diseño gráfico digital. Los diseños se realizaron utilizando Moqups, lo que permite el uso de elementos independientes y da cierta libertad en los aspectos técnicos del diseño.

Figura 18

Desarrollo del diseño de la interfaz Inicio de sesión con Moqups.



Se tuvo que crear una interfaz de usuario digital para evaluar con mayor precisión el diseño final. Además de proporcionar la capacidad de utilizar medidas precisas como base para un mayor desarrollo del diseño.

Los colores utilizados para la aplicación se determinan de acuerdo al logotipo de la misma. Esto le permite mantener el mismo entorno en todo el concepto, proporcionar su propio entorno o identificar la forma en que el usuario se encuentra en el entorno de la aplicación.

Figura 19

Diseño de interfaz de Home Administrador

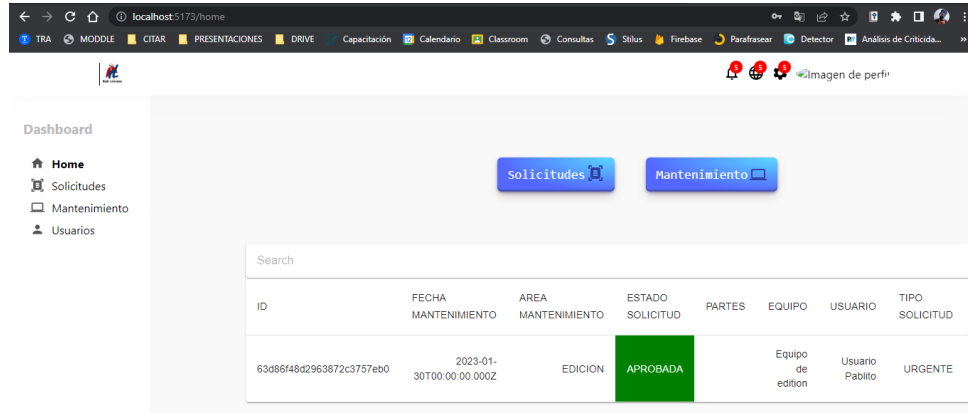
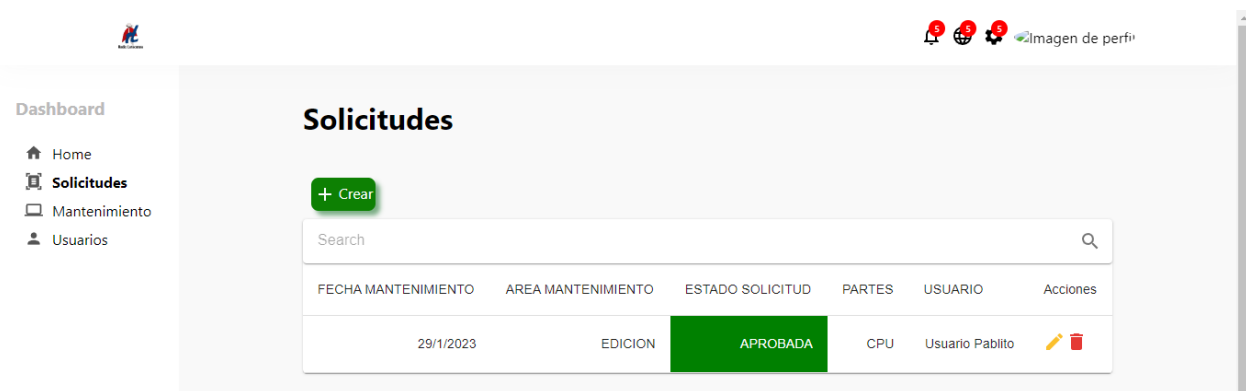


Figura 20

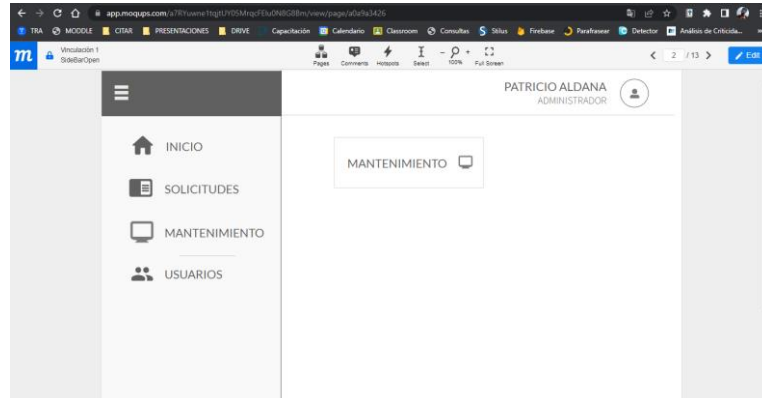
Diseño de Solicitudes



Maqueta. Para el desarrollo de la versión prototipo se utilizó Moqups como herramienta que permitía “componer” cada imagen de una manera determinada, relacionarlas con un punto determinado de la aplicación y permitir realizar pruebas de experiencia de usuario.

Figura 21

Progreso del diseño de interfaz con Moqups.



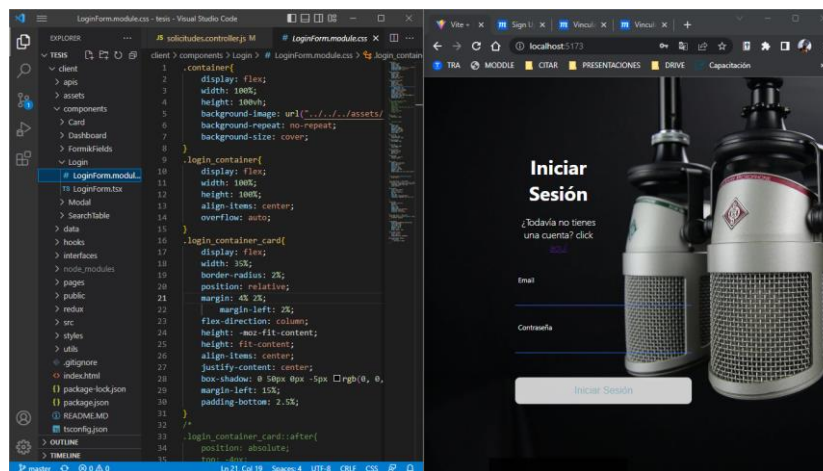
Implementación

Desarrollo de servicio de negocio

La codificación de la interfaz se desarrolla utilizando el lenguaje JavaScript y el diseño utiliza el cliente-servidor NodeJS con la biblioteca ReactJS.

Figura 22

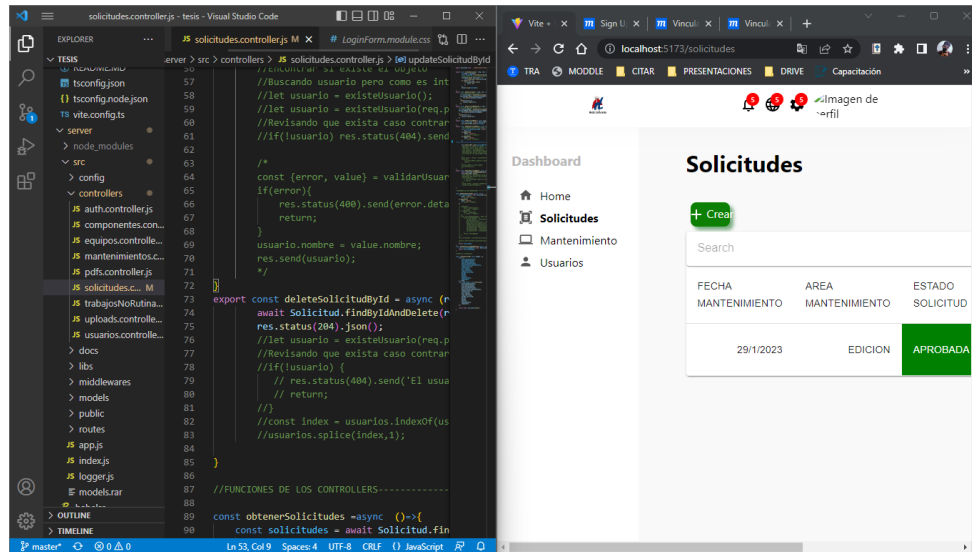
Maquetado de bocetos Inicio de Sesión.



Nota. Codificación y simbolización por medio de lenguaje de programación para diseños.

Figura 23

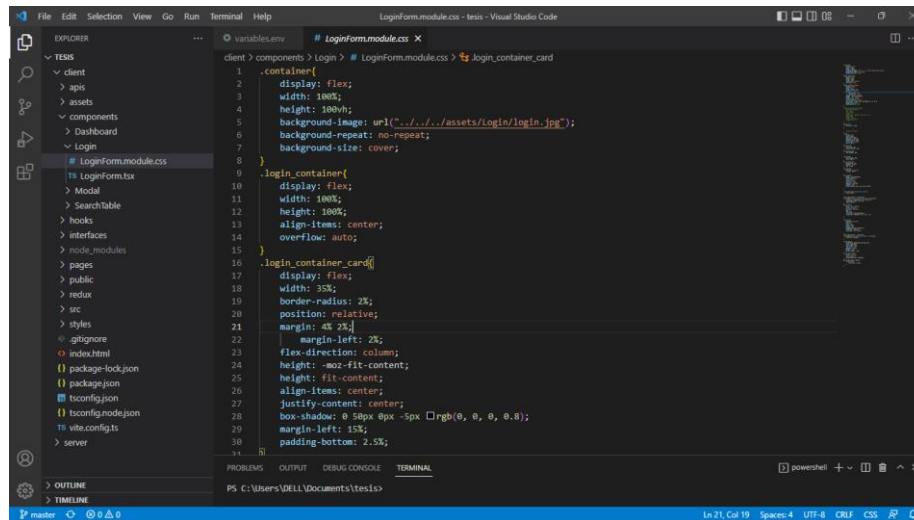
Maquetado de Interfaz de Solicitudes



Nota. Codificación y simbolización por medio de lenguaje de programación para diseños.

Figura 24

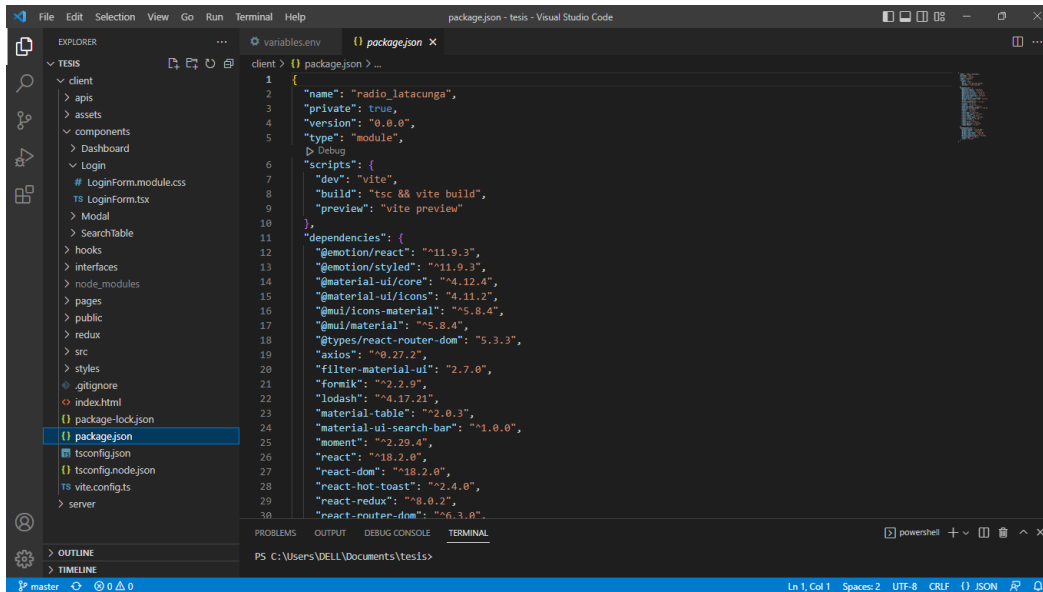
Codificación utilizando CSS.



Nota. Codificación y simbolización por medio del lenguaje de programación para diseños.

Figura 25

Maquetación mediante JSX



```

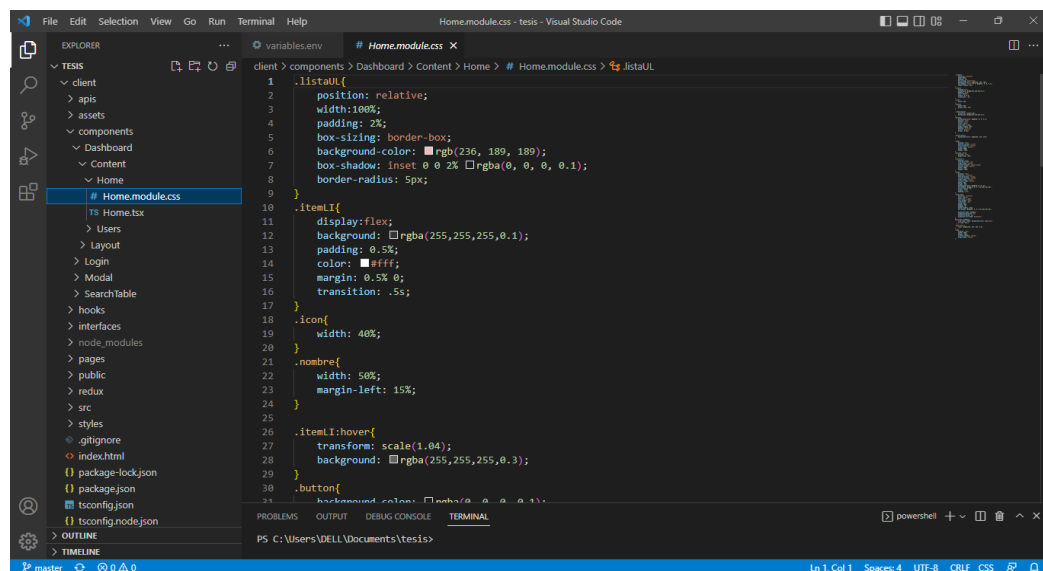
1 {
2   "name": "radio_latacunga",
3   "private": true,
4   "version": "0.0.0",
5   "type": "module",
6   "scripts": {
7     "dev": "vite",
8     "build": "tsc && vite build",
9     "preview": "vite preview"
10  },
11  "dependencies": {
12    "@emotion/react": "^11.9.3",
13    "@emotion/styled": "^11.9.3",
14    "@material-ui/core": "^4.12.4",
15    "@material-ui/icons": "4.11.2",
16    "@mui/icons-material": "^5.8.4",
17    "@mui/material": "^5.8.4",
18    "@types/react-router-dom": "5.3.3",
19    "axios": "^0.27.2",
20    "filter-material-ui": "2.7.0",
21    "formik": "^2.2.9",
22    "lodash": "^4.17.21",
23    "material-table": "^2.0.3",
24    "material-ui-search-bar": "^1.0.0",
25    "moment": "^2.29.4",
26    "react": "^18.2.0",
27    "react-dom": "^18.2.0",
28    "react-hot-toast": "^2.4.0",
29    "react-redux": "^8.0.2",
30    "react-router-dom": "^6.3.0"
  }
}

```

Nota. Codificación y simbolización por medio del lenguaje de programación para diseños.

Figura 26

Maquetado de Home CSS



```

1  .listaUl{
2    position: relative;
3    width:100%;
4    padding: 2%;
5    box-sizing: border-box;
6    background-color: #rgb(236, 189, 189);
7    box-shadow: inset 0 0 2% #rgba(0, 0, 0, 0.1);
8    border-radius: 5px;
9  }
10
11  .itemLi{
12    display:flex;
13    background: #rgba(255,255,255,0.1);
14    padding: 0.5%;
15    color: #fff;
16    margin: 0.5% 0;
17    transition: .5s;
18  }
19  .icon{
20    width: 40%;
21  }
22  .nombre{
23    width: 50%;
24    margin-left: 15%;
25  }
26  .itemLi:hover{
27    transform: scale(1.04);
28    background: #rgba(255,255,255,0.3);
29  }
30  .button{
31    background-color: #rgb(0, 0, 0, 0.1);
32  }

```

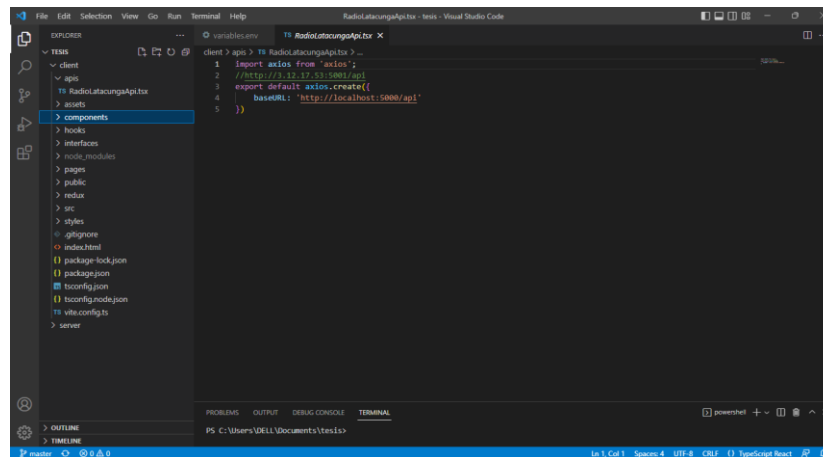
Nota. Codificación y simbolización por medio del lenguaje de programación para diseños.

Integración con la plataforma

La conexión a los microservicios se desarrolla utilizando una biblioteca NodeJS creada que realiza solicitudes a la API para comunicarse con el cliente, y esta biblioteca es Axios, que es un cliente REST de la API.

Figura

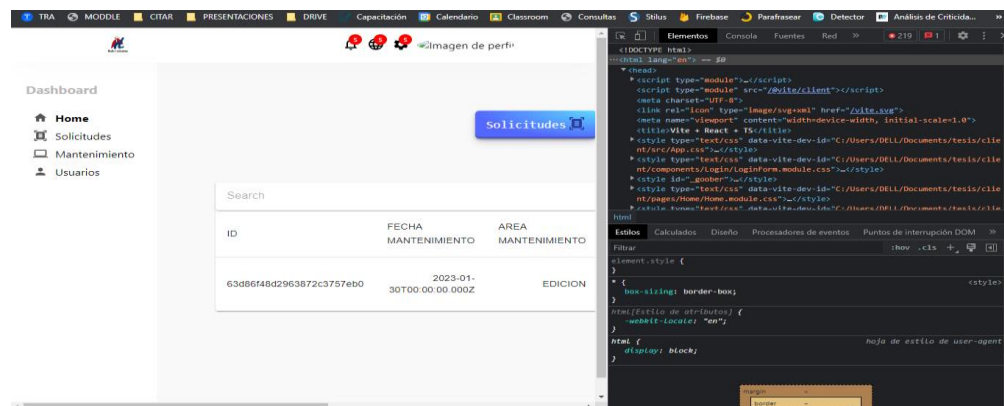
Conexiones a las APIs mediante Axios



Nota. Codificación de las correspondencias a las APIs por medio de Axios.

Figura 27

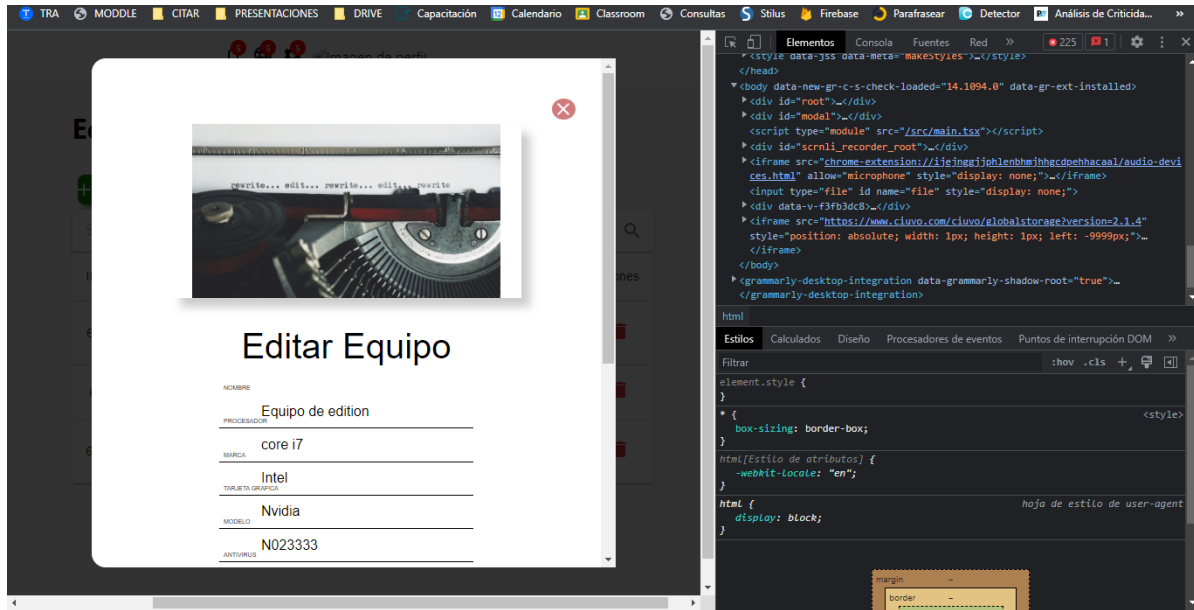
Validación de API



Nota. Codificación de las correspondencias a las APIs por medio de Axios.

Figura 28

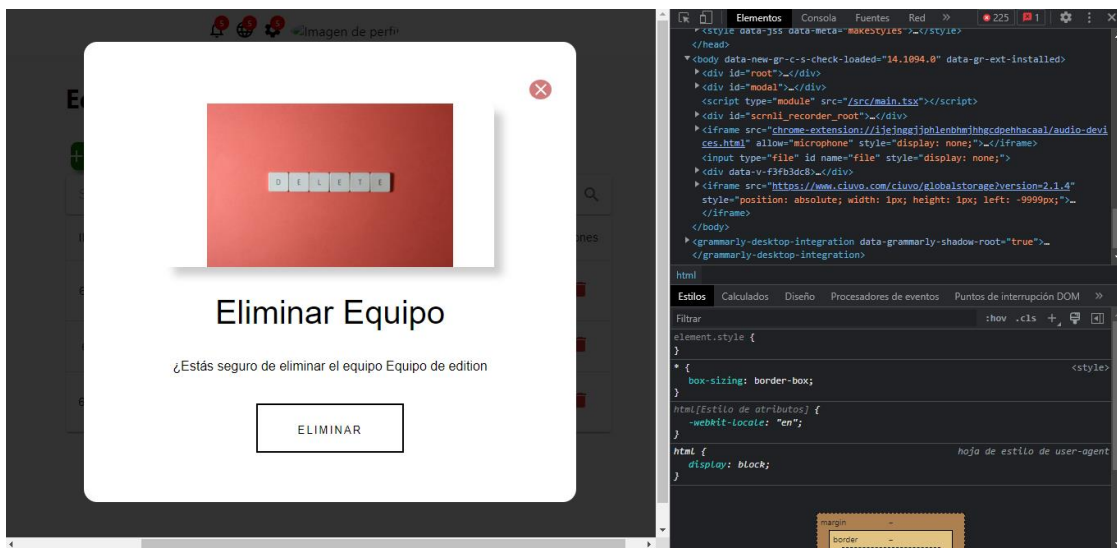
Codificación de las correspondencias o conexiones.



Nota. Codificación de las correspondencias a las APIs por medio de Axios.

Figura 29

Validación mediante API



Nota. Codificación de las correspondencias a las APIs por medio de Axios.

Clonación del repositorio. El repositorio ya hecho del aplicativo web se clona en la instancia del servidor.

Figura 32

Clonación del repositorio en la instancia

```
root@ip-172-31-31-194:/home/ubuntu/radio_latacunga_server# npm install
npm WARN deprecated crypto@1.0.1: This package is no longer supported. It's now a built-in Node module. If you've depended on crypto, you should switch to the one that's built-in.
npm WARN deprecated @hapi/pinpoint@2.0.0: Moved to 'npm install @sideway/pinpoint'
npm WARN deprecated @hapi/formula@2.0.0: Moved to 'npm install @sideway/formula'
npm WARN deprecated @hapi/address@4.1.0: Moved to 'npm install @sideway/address'
npm WARN deprecated querystring@0.2.0: The querystring API is considered Legacy. new code should use the URLSearchParams API instead.
npm WARN deprecated uuid@3.3.2: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
npm WARN deprecated @hapi/joi@17.1.1: Switch to 'npm install joi'
(1) reify:babel/parser: GET 200 https://registry.npmjs.org/@babel/parser/-/parser-7.16.6.tgz 4797ms (cache miss)
```

Nota. La clonación del repositorio se debe hacer ya una vez finalizada el aplicativo como también seleccionado el servidor.

Instalación de dependencias. En el servidor que se seleccionó para que se almacene el aplicativo web se instala cada dependencia para un correcto alojamiento y funcionamiento del aplicativo web.

Figura 33

Instalación de las dependencias en el servidor

```
root@ip-172-31-31-194:/home/ubuntu# git clone https://github.com/GrayBismuthMaster/radio_latacunga_server
Cloning into 'radio_latacunga_server'...
remote: Enumerating objects: 116, done.
remote: Counting objects: 100% (116/116), done.
remote: Compressing objects: 100% (79/79), done.
remote: Total 116 (delta 59), reused 88 (delta 31), pack-reused 0
Receiving objects: 100% (116/116), 1.67 MiB | 26.75 MiB/s, done.
Resolving deltas: 100% (59/59), done.
root@ip-172-31-31-194:/home/ubuntu#
```

Inicio del servidor. Ya una vez instalada cada una de las dependencias se procede a iniciar el servidor en el cual se alojará el aplicativo web teniendo en consideración los datos que alojara el aplicativo web y los datos que se ingresaran y se descargan en el aplicativo web.

Figura 34

Inicio del servidor Amazon Web Services

```

root@ip-172-31-31-194:/home/ubuntu/radio_latacunga_server# pm2 list

```

id	name	namespace	version	mode	pid	uptime	♻	status	cpu	mem	user	watching
0	Radio Latacunga Api	default	N/A	fork	2102	78s	1	online	0%	63.7mb	root	disabled

```

[PM2] [WARN] Current process list is not synchronized with saved list. App ELS Server Crissol server Pointi Server Crissol Cliente Pointi Cliente differs. Type 'pm2 sav

```

Nota. Inicio del servidor Amazon Web Services en estado online.

Creación de zona alojada. Para la creación se hace un registro de creación rápida en la cual se selecciona el tipo de registro lo cual servirá para identificar el aplicativo web y se obtiene una dirección Ip como también se registra el dominio con el cual estará registrado el aplicativo web.

Figura 35

Creación de zona alojada en el servidor.

Crear registro [Info](#)

Registro de creación rápida [Cambiar al asistente](#)

▼ Registro 1 [Eliminar](#)

Nombre del registro [Info](#) radio-latacunga.com

Mantenga el espacio en blanco para crear un registro para el dominio raíz.

Tipo de registro [Info](#)

Alias

Valor [Info](#)

Introduzca varios valores en líneas separadas.

TTI (segundos) [Info](#) Política de direccionamiento [Info](#)

Pruebas Unitarias

Aplicación Web. En base a las pruebas realizadas, las pruebas muestran que la aplicación web cumple con los aspectos técnicos requeridos.

Tabla 10

El resultado de cargar la aplicación y cargar los componentes en el navegador

ID Caso de prueba	Descripción de la prueba	Datos de prueba	Pasos a ejecutar	Resultado Esperado	Resultado real	Contraseña errónea
CP01	Verificación la aplicación para ver cómo funciona en Google Chrome	Para realizar la prueba, ingresa al siguiente link:	<ul style="list-style-type: none"> • Ingresar al URL • Visibilizar su renderización • Comprobar contenido 	Test React debe obedecer a cada componente para renderizar con éxito	Test React renderiza sin problemas, estos componentes defectuosos se han solucionado.	Pasado
CP02	Verificación la aplicación para ver cómo funciona en Edge	Para realizar la prueba, ingresa al siguiente link:	<ul style="list-style-type: none"> • Ingresar al URL • Visibilizar su renderización • Comprobar contenido 	Renderiza los componentes contruidos sin errores	El navegador representa cada elemento de la interfaz junto con los estilos definidos sin ningún problema.	Pasado

ID	Descripción de la prueba	Datos de prueba	Pasos a ejecutar	Resultado Esperado	Resultado real	Contraseña errónea
CP03	Verificación la aplicación para ver cómo funciona en Firefox	Para realizar la prueba, ingresa al siguiente link:	<ul style="list-style-type: none"> • Ingresar al URL • Visibilizar su renderización • Comprobar contenido 	Renderiza los componentes construidos sin errores	El navegador representa cada elemento de la interfaz junto con los estilos definidos sin ningún problema.	Pasado
CP04	Renderizado de componentes	Componente a renderizar	<p>Ejecutar el comando de renderizado</p> <p>Verificar los datos obtenidos</p>	Renderiza los componentes construidos sin errores	El navegador representa cada elemento de la interfaz junto con los estilos definidos sin ningún problema.	Pasado

Nota. Como se puede ver, los datos obtenidos se acercan a los resultados esperados, se logró la correcta ejecución de la interfaz sin mayores problemas en todos los navegadores. Es posible que los componentes se hayan tergiversado, pero se han reparado correctamente.

Microservicios. Para obtener una respuesta inmediata a la API, se debe reducir la cantidad de solicitudes a la base de datos, por lo que se está examinando el tiempo de solicitud y los métodos para mejorar esta comunicación. Estos datos se presentan en la tabla 11.

Tabla 11

Obtención resultados con tiempo de consulta optimizado en microservicios.

ID Caso de prueba	Descripción de la prueba	Datos de prueba	Pasos a ejecutar	Resultado Esperado	Resultado real	Contraseña errónea
CP05	Tiempo de consulta por punto final	Tiempo creado por Postman	<ul style="list-style-type: none"> Ejecutar solicitudes para cada punto final en Postman Consulta el tiempo de resultado en la aplicación Postman 	En las pruebas locales, el tiempo de compromiso es inferior a 1 segundo.	Las opiniones consultivas brindadas superaron las expectativas y fueron corregidas oportunamente.	Pasado

Pruebas de integración. Cuando se establece la conexión entre el microservicio y la aplicación, se consideran dos modos de prueba: local y remoto.

Tabla 12

Resultados de las pruebas de integración de microservicios y aplicaciones web

ID Caso de prueba	Descripción de la prueba	Datos de prueba	Pasos a ejecutar	Resultado Esperado	Resultado real	Contraseña errónea
CP06	Comprobar el tiempo de respuesta de los microservicios	Tiempo de adquisición de datos y muestreo	<ul style="list-style-type: none"> • Cargar sintaxis de depuración en el código de React • Observar resultados en navegador 	Obtenga respuestas en menos de 1,5 segundos y renderice en menos de 1 segundo.	Se logró un tiempo de respuesta promedio de 2 segundos.	Pasado
CP07	Verificar el tiempo de respuesta usando API Gateway	Tiempo de adquisición de datos y muestreo	<ul style="list-style-type: none"> • Cargar sintaxis de depuración en el código de React • Observar resultados en navegador 	El tiempo total de reacción no debe exceder los 3 segundos.	El valor máximo de consulta se logró con un resultado de 2,83 segundos.	Pasado

ID Caso de prueba	Descripción de la prueba	Datos de prueba	Pasos a ejecutar	Resultado Esperado	Resultado real	Contraseña errónea
CP08	Obtener tiempo de carga externamente	Tiempo de adquisición de datos y muestreo	<ul style="list-style-type: none"> • Ingresar a la URL • Verificar los registros de los microservicios y aplicación web 	El tiempo de respuesta del microservicio debe ser local	El tiempo de respuesta es de 0,6 segundos para el microservicio y de 1 segundo para la aplicación web.	Pasado

Nota. Las validaciones locales, semilocales y remotas brindan un proceso de retroalimentación para obtener mejores resultados en la implementación final debido al nivel de procesamiento de los servicios en la nube.

Pruebas de rendimiento. El rendimiento se logra mediante la optimización del código y las pruebas para verificar estos resultados, además de las aplicaciones que verifican esto mediante la simulación de solicitudes de mantenimiento simultáneas (como GTmetrix).

Tabla 13*Resultados de la prueba de rendimiento*

ID Caso de prueba	Descripción de la prueba	Datos de prueba	Pasos a ejecutar	Resultado Esperado	Resultado real	Contraseña errónea
CP10	Validar CSS y Javascript no utilizado	Muestreo de CSS y Javascript no utilizado	<ul style="list-style-type: none"> • Ingresar a GTmetrix • Cargar el enlace del aplicativo web • Validare datos resultantes 	La herramienta debe informar que no puede encontrar CSS y Javascript no utilizados	La aplicación detectó que se eliminaron CSS y Javascript no utilizados	Pasado
CP11	Validar los procesos que realiza	Cantidad de solicitudes simultáneas que se realizan a los microservicios	<ul style="list-style-type: none"> • Ingresar a GTmetrix • Cargar el enlace del aplicativo web • Validar datos resultantes 	Manejar la menor cantidad de solicitudes de microservicio simultáneas	Se enviaron más de 3 solicitudes al mismo tiempo y se corrigieron a tiempo	Pasado

Nota. Seguir las recomendaciones de GTmetrix de la aplicación te permite mejorar y corregir aspectos técnicos que pueden ralentizar la aplicación.

Capítulo IV

Conclusiones y Recomendaciones

Conclusiones

La aplicación web está diseñada para facilitar la automatización de la información de la emisora Latacunga, reunir todos los requisitos necesarios y cubrir todas las necesidades deseadas. Se creó una interfaz sencilla e intuitiva para que el usuario pueda trabajar rápidamente con la aplicación, nos mostrará los datos correctos donde se espera el mantenimiento, el mantenimiento realizado y las solicitudes de mantenimiento atendidas, creando herramientas para ayudar en el trabajo.

En conclusión, se implementó una herramienta informática que permitirá a la Radio Latacunga la recolección de cada uno de los datos de mantenimientos de los equipos TICS además de facilitar la construcción de un nuevo plan de mantenimiento y así facilitar el registro y organización de información, evitar así fallos en los equipos o sistemas de la radio.

Como se ha podido observar el aplicativo web se desarrolló para ejecutar un plan de mantenimiento preventivo y correctivo ya sea de software y hardware en los equipos TIC'S en la radio Latacunga además es capaz de enlistar y gestionar solicitudes de mantenimiento.

Recomendaciones

Es recomendable que el personal de la radio Latacunga tenga conocimiento acerca de la utilización del aplicativo web de tal manera que, si llegara a existir algún fallo urgente en el equipo informático, el usuario pueda solicitar un mantenimiento mediante el aplicativo web.

Es necesario realizar un buen manejo del aplicativo web para que de esta manera se pueda llevar el registro adecuado de cada plan de mantenimiento ejecutado en los equipos TICS de la radio Latacunga y evitar cualquier inconveniente con los micros servicios de la aplicación.

Es importante tomar nota de futuros requerimientos que se presenten con el tiempo para de esta manera implementarlos en el futuro mejorando así el rendimiento y posiblemente creando nuevas funcionalidades de acuerdo con el crecimiento de la radio Latacunga.

Glosario de Términos

SCRUM: Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos («Qué es SCRUM», 2008).

Aplicaciones Web: Una aplicación web o "aplicación web" es un programa. programa que se ejecuta en un servidor web. A diferencia de las aplicaciones de escritorio tradicionales, que son lanzadas por su sistema operativo, se debe acceder a las aplicaciones web a través de un navegador web (Definición de aplicación web, s. f.)

React: (también llamada React.js o ReactJS) es una biblioteca JavaScript de código abierto para crear interfaces de usuario con el objetivo de animar al desarrollo de aplicaciones en una sola página ((3) DesarrolladorSoft - Publicaciones | Facebook, s. f.)

JavaScript: es una especie de lenguaje de programación ligera, interpretado por la mayoría de los navegadores y que les proporciona a las páginas web, efectos y funciones complementarias a las consideradas como estándar HTML (¿Qué es JavaScript?, s. f.)

Bibliografía

Atlassian. (s.f.). Atlassian Bitbucket. Obtenido de Que es el control de versiones:

<https://www.atlassian.com/es/git/tutorials/what-is-version-control>

Brocca, J. C., & Camasaqui, R. (2005). Las Licencias de Software Desde las Perspectivas del Usuario Final. Pilquen, 10.

Cabello Garcia, J. M. (2015). Almacenamiento de la información e introducción a los SGBD.

Andalucía, España: IC Editorial. Obtenido de

<https://elibro.net/es/ereader/espe/44147?page=45>

Campderrich Falgueras, B. (2003). Ingeniería de Software. Catalunya: UOC.

Casado Iglesias, C. (2015). Entornos de Desarrollo. Madrid, España: RA-MA.SA Editorial.

Obtenido de <https://elibro.net/es/ereader/espe/62495?page=45>

Moreno, J. C., & Ramos, A. F. (2014). Administración Software de un Sistema Informático.

Madrid: RA-MA. SA Editorial y Publicaciones.

Ordax Cassá, J. M. (2013). Programación web en Java. Madrid, España: Gobierno de España.

Pressman, R. (2010). Ingeniería del Software Un Enfoque Práctico. México: McGRAW-HILL .

Salazar López, B. (2020). Ingeniería Industrial. Obtenido de Ingeniería Industrial:

<https://www.ingenieriaindustrialonline.com/gestion-de-almacenes/que-es-la-gestion-de-almacenes/>

Vela, B. (2003). Una Metodología basada en Modelos para el Desarrollo de la Dimensión

Estructural de Sistemas de Información Web. (Tesis Doctoral). Universidad Rey Juan

Carlos, España.

Concepto.de. (2017). Concepto.de. Obtenido de <https://concepto.de/fuente-de-alimentacion/>

OptimusPC. (8 de Agosto de 2017). OptimusPC. Obtenido de

<https://optimuspc.com.ar/producto/fuente-noga-500w-atx/>

Significados. (28 de Enero de 2015). Significados. Obtenido de

<https://www.significados.com/mantenimiento-preventivo/>

Tecnología Fácil. (2015). Tecnología Física. Obtenido de [https://tecnologia-facil.com/que-](https://tecnologia-facil.com/que-es-que-es-una-fuente-de-alimentacion-de-pc/)

[es-que-es-una-fuente-de-alimentacion-de-pc/](https://tecnologia-facil.com/que-es-que-es-una-fuente-de-alimentacion-de-pc/)

Visión Tecnológica. (s.f.). Visión Tecnológica. Obtenido de

<https://www.visiontecnologica.net/mantenimiento-preventivo-computadoras/>

Qué es SCRUM. (2008, agosto 4). Proyectos Ágiles. <https://proyectosagiles.org/que-es-scrum/>

¿Qué es JavaScript?» Su Definición y Significado [2021]. (s. f.). Concepto de - Definición de.

Recuperado 9 de marzo de 2021, de [//conceptodefinicion.de/javascript/](http://conceptodefinicion.de/javascript/)

Eguíluz Pérez Javier. (s. f.). Introducción a JavaScript. 140.

elastic. (s. f.-a). ¿Qué es Elasticsearch? | Elastic. Recuperado 17 de julio de 2022, de

<https://www.elastic.co/es/what-is/elasticsearch>

¿Qué es una base de datos? (2014). Oracle.com. [https://www.oracle.com/mx/database/what-is-](https://www.oracle.com/mx/database/what-is-database/)

[database/](https://www.oracle.com/mx/database/what-is-database/)

Jiménez Calderón, Mario. (2020). idUS - Ingeniería de pruebas: Aplicación, alcance y

rendimiento en un banco de test para una aplicación cliente-servidor. Ingeniería de

pruebas: aplicación, alcance y rendimiento en un banco de test para una aplicación

cliente-servidor. <https://idus.us.es/handle/11441/104616>

Marzal Varó, A., & Gracia Luengo, I. (2009). Introducción a la programación con Python.

Universitat Jaume I. <http://repositori.uji.es/xmlui/handle/10234/24305>

Pastor, J. J. C. (2016). Motores de búsqueda y derechos de autor: Infracción y responsabilidad.

Aranzadi Thomson Reuters.

Puciarelli, L. (2020). Node JS - Vol. 1: Instalación - Arquitectura - node y npm. RedUsers.

Rubiales Gómez Mario. (2021). CURSO DE DESARROLLO WEB: HTML, CSS Y JAVASCRIPT.

En Manual Imprescindible (2021.a ed., p. 345). ANAYA.

https://anayamultimedia.es/primer_capitulo/curso-de-desarrollo-web-html-css-y-javascript-edicion-2021.pdf

Ryan Wynn. (2016). Monstache. <https://rwynn.github.io/monstache-site/>

Vértice, E. (2009). Diseño básico de páginas web en HTML. Editorial Vértice.

¿Qué es una base de datos relacional? (2014). Oracle.com.

<https://www.oracle.com/ar/database/what-is-a-relational-database/>

Bases de datos no relacionales | Bases de datos de gráficos | AWS. (2023). Amazon Web

Services, Inc. <https://aws.amazon.com/es/nosql/>

Lucero, M. (2023). 2.4. Estructura de la base de datos - Maricruz Lucero Lugo Coutiño.

Google.com. <https://sites.google.com/site/mariluz20188/2-4-estructura-de-la-base-de-datos>

(S/f). Uji.es. Recuperado el 24 de enero de 2023, de

<http://www3.uji.es/~mmarques/f47/teoria/tema3.pdf>

Modelado de datos conceptual. (s/f). Erwin.com. Recuperado el 24 de enero de 2023, de

<https://www.erwin.com/mx-es/solutions/data-modeling/conceptual.aspx>

IBM Documentation. (2021, marzo 8). Ibm.com.

<https://www.ibm.com/docs/es/ida/9.1.2?topic=modeling-logical-data-models>

IBM Documentation. (2021, Marzo 17). Ibm.com. [https://www.ibm.com/docs/es/data-](https://www.ibm.com/docs/es/data-studio/4.1.1?topic=modeling-physical-data-models)

[studio/4.1.1?topic=modeling-physical-data-models](https://www.ibm.com/docs/es/data-studio/4.1.1?topic=modeling-physical-data-models)

Arquitectura de software: Qué es y qué tipos existen. (2022, Agosto 24). OpenWebinars.net.

<https://openwebinars.net/blog/arquitectura-de-software-que-es-y-que-tipos-existen/>

Qué es Frontend y Backend: diferencias y características - Platzi. (2023). Platzi.

[https://platzi.com/blog/que-es-frontend-y-](https://platzi.com/blog/que-es-frontend-y-backend/#:~:text=Frontend%20es%20la%20parte%20de,corre%20del%20lado%20del%20servido)

[backend/#:~:text=Frontend%20es%20la%20parte%20de,corre%20del%20lado%20del%20servido](https://platzi.com/blog/que-es-frontend-y-backend/#:~:text=Frontend%20es%20la%20parte%20de,corre%20del%20lado%20del%20servido)

Anexos

Documentos:

Carpeta de anexo

Tesis en word pdf

Diaposti word pdf

Resuemen

Antiplagio

Hojas de firma

Caja guresa

Protectos de johas de plastico