

DEPARTAMENTO DE INFORMÁTICA
CARRERA: INGENIERÍA INFORMÁTICA

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO INFORMÁTICO

Migración de software relacionados con el cálculo del potencial del biogás en entidades que generan residuos
Software migration related to the calculation of biogas potential

Autor: Angélica María Rivero Abreu

Tutor: Dr.C. Luis Quintero Domínguez
MSc. Jarvin Antón Vargas

Este documento es Propiedad Patrimonial de la Universidad de Sancti Spíritus «José Martí Pérez», y se encuentra depositado en los fondos del Centro de Recursos para el Aprendizaje y la Investigación «Raúl Ferrer Pérez», subordinado a la Dirección General de Desarrollo 3 de la mencionada casa de altos estudios.

Se autoriza su utilización bajo la licencia siguiente:

Atribución- No Comercial- Compartir Igual



Para cualquier información, contacte con:

Centro de Recursos para el Aprendizaje y la Investigación “Raúl Ferrer Pérez”. Comandante Manuel Fajardo s/n, esquina a Cuartel, Olivos 1. Sancti Spíritus. Cuba. CP. 60100

Teléfono: **41-334968**

Me gustaría dedicar este proyecto de tesis a mis padres, que si no fuese por su apoyo y dedicación no sería la persona que soy hoy...

AGRADECIMIENTOS

Me gustaría agradecer a mis padres, por tenerme paciencia y ayudarme todo lo que han podido y en lo que no, el apoyo de que pase lo que pase van a estar ahí conmigo,

A mis tutores, que me han apoyado en todo lo que he necesitado y han estado ahí para ayudarme,

En especial a mis compañeros de aula, porque han sido mucho más que eso en estos cuatro años y me han hecho sentir en familia y cualquier duda que tenga me pueden ayudar,

A mi familia y amigos en general, que siempre se han brindado para ayudar en cualquier momento.

RESUMEN

El biogás es un gas combustible que se genera por la acción de microorganismos en ausencia de oxígeno. Constituye una fuente de energía renovable no contaminante que se utiliza con éxito en varios países del mundo. Calcular su potencial es muy importante porque brinda información acerca de la cantidad de biogás que genera un tipo de residuo. En nuestro país se aprecia un significativo potencial para la producción y utilización del biogás, basado en la diversidad y el volumen de residuos contaminantes. Con la finalidad de facilitar este cálculo de potencial fueron creados dos sistemas para el Cento de Estudios de la Energía y los Procesos Industriales de la Universidad de Sancti Spíritus, sistemas que por separado no cumplen con las expectativas del cliente porque tienen varios elementos similares, pero que de unirlos en un sólo sistema sí estarían cumpliendo las expectativas, además de la necesidad de agregar nuevas funcionalidades. Se decide entonces la migración de estos dos sistemas para crear una única aplicación utilizando tecnologías basadas en el ambiente Node.JS (Strapi como backend, Nuxt.JS como frontend usando Vuetify.JS), donde se mejoren las funcionalidades agregando además nuevos residuos que se calculan de manera diferente, obteniéndose un sistema escalable, organizado y con la información centralizada.

Palabras clave: migración, Node.JS, Strapi, Nuxt.JS, API, Vuetify.JS

ABSTRACT

Biogas is a combustible gas generated by the action of microorganisms in the absence of oxygen. It is a non-polluting renewable energy source that is successfully used in several countries around the world. Calculating its potential is very important because it provides information about the amount of biogas generated by a type of waste. In our country, a significant potential for biogas production and utilization is appreciated, based on the diversity and volume of polluting waste. In order to facilitate this potential calculation, two systems were created for the Center for Energy and Industrial Processes Studies of the University of Sancti Spíritus, systems that separately do not meet the client's expectations because they have several similar elements, but if they were to be united in a single system, they would meet the expectations, in addition to the need to add new functionalities. It is then decided to migrate these two systems to create a single application using technologies based on the Node.JS environment (Strapi as backend, Nuxt.JS as frontend using Vuetify.JS), where the functionalities are improved by adding new residues that are calculated in a different way, obtaining a scalable, organized and centralized information system.

Summary Keywords: migration, Node.JS, Strapi, Nuxt.JS, API, Vuetify.JS

ÍNDICE

FIGURAS	IX
TABLAS	XI
INTRODUCCIÓN	1
1. Fundamentos para la migración de dos sistemas informáticos	5
1.1. Biogás, producción y utilización	5
1.1.1. Cálculo del potencial implementable del biogás	8
1.2. Geolocalización	10
1.3. Migración	10
1.4. Sistema Heredado	13
1.4.1. Tipo de migraciones de software heredados	14
1.4.2. Metodología MADIISH para la migración de sistemas heredados . .	14
1.5. Herramientas Informáticas	17
1.6. Conclusiones parciales	21
2. Planificación, diseño e implementación de la nueva aplicación web	22
2.1. Factores a tener en cuenta para el desarrollo de la migración	22

2.2. Requisitos	26
2.2.1. Requisitos funcionales	26
2.2.2. Requisitos no funcionales	31
2.2.3. Requisitos de Software	31
2.3. Buenas prácticas para el desarrollo del nuevo software	32
2.4. Fases del ciclo de migración para el diseño del sistema	32
2.4.1. Fase Cero	32
2.4.2. Fase 1: Análisis de procesos de negocio	34
2.4.3. Fase 2: Modelado, programación y generación del prototipo	36
2.5. Conclusiones parciales	43
3. Implementación y pruebas	50
3.1. Datos generales del proyecto	50
3.2. Fases del ciclo de migración para el desarrollo y la implementación del sistema	51
3.2.1. Fase 3: Pruebas y resultados	51
3.2.2. Fase 4: Impementación	55
3.2.3. Fase 5: Migración continua y documentación	64
3.2.4. Fase T: Elaboración de la documentación final	64
3.3. Conclusiones parciales	64
CONCLUSIONES	65
RECOMENDACIONES	67
REFERENCIAS	68
ANEXOS	69

A. Estadísticas	70
B. Diagramas de la Ingeniería del Software	72
C. Vistas de las aplicaciones	74
D. Manuales de Usuario	76

FIGURAS

2.1. Diagrama de entradas y salidas del inicio de sesión en el sistema base	35
2.2. Diagrama de entradas y salidas de operaciones del sistema. Agregar una entidad	36
2.3. Diagrama entidad relación del diseño de la base de datos del sistema uno: Agropecuario y RSU (residuos sólidos urbanos)	37
2.4. Diagrama entidad relación del diseño de la base de datos del sistema dos: Agroindustrial	38
3.1. Secuencia de pasos realizados en Selenium de manera exitosa para añadir una nueva entidad en el sistema, pasando primero por el inicio de sesión	57
3.2. Pruebas de rendimiento realizadas al sistema para los métodos get	58
3.3. Pruebas de rendimiento para el método delete	59
3.4. Pruebas de rendimiento para el método post	59
3.5. Prueba de rendimiento y funcionalidad realizada a los filtros del reporte . . .	60
3.6. Prueba de rendimiento realizada a los filtros del reporte por tipo de entidad . .	60
3.7. Prueba de funcionalidad realizada a la interfaz de geolocalización	61
3.8. Agregando una entidad al nuevo sistema	62
3.9. Filtrados en la interfaz con el paginador <i>server side</i> a elemento por tipo de residuo	63
3.10. Documento excel con todos los elementos que encontró la base de datos a partir de esos filtros	63
A.1. Índice TIOBE	71
A.2. Escala anual de Github sobre el uso de lenguajes de programación en los últi- mos años	71

B.1. Diagrama físico de la nueva base de datos	73
C.1. Geolocalización de las aplicaciones antiguas	75
C.2. Geolocalización de la nueva aplicación	75
D.1. Pasos en la consola para crear el proyecto nuxt	77

TABLAS

2.1. Características de los usuarios del nuevo sistema	26
2.2. Actividades que se realizan en la fase cero de MAADISH	33
2.3. Descomposición del sistema en sus diferentes sistemas	34
2.4. Actividades que se realiza en la fase 1	35
2.5. Actividades de la fase dos de MAADISH	37
2.6. Modelo entidad relación de la nueva base de datos a ser utilizada, integrando los modelos de bases de datos de los sistemas anteriores y agrupando entidades	39
2.7. Diseño de la interfaz	40
2.8. Panel Antiguo de las aplicaciones	40
2.9. Panel de la aplicación actual	41
2.10. <i>Login</i> antiguo	41
2.11. <i>Login</i> de la aplicación actual	42
2.12. Gestión de un residuo en la aplicación actual	42
2.13. Gestión de un residuo en los sistemas anteriores	43
2.14. Agregar un residuo en los sistemas anteriores/ Agregar un residuo en la aplicación actual	44
2.15. Potenciales en las aplicaciones anteriores	45
2.16. Vista de los potenciales en la aplicación actual	45
2.17. Perfil <i>frontend</i> y <i>backend</i> para la gestión de un residuo	46
2.18. Perfil <i>backend</i> y <i>frontend</i> para la vista del mapa	47
2.19. Perfil <i>backend</i> y <i>frontend</i> para el inicio de sesión	48
2.20. Menú lateral del <i>frontend</i>	49
3.1. Actividades de la fase 3 de MAADISH	52

TABLAS

3.2. Formato de pruebas para el <i>login</i>	53
3.3. Formato plan de pruebas para el registro de un nuevo usuario	54
3.4. Formato plan de pruebas para agregar una nueva entidad al sistema	56
3.5. Organización del nuevo sistema en cuanto a módulos y servidores	57

INTRODUCCIÓN

El biogás es un gas combustible que se genera por la acción de microorganismos en ausencia de oxígeno. Este proceso se conoce como digestión anaeróbica y es realizado por distintos tipos de bacterias. Constituye una fuente de energía renovable no contaminante que se utiliza con éxito en varios países del mundo. Existe una amplia variedad de biomasa que puede ser transformada en biogás, desde residuos agrícolas provenientes de la cosecha de plantaciones anuales, lodos de plantas de tratamiento de aguas contaminadas con desechos orgánicos o inorgánicos; de rellenos sanitarios, y desechos orgánicos del sector ganadero, especialmente de vacuno y porcino con capacidad de fermentarse (Sánchez, 2016).

En nuestro país se aprecia un significativo potencial para la producción y utilización del biogás, basado en la diversidad y el volumen de residuos contaminantes que generan los sectores agropecuario y agroindustrial. Sin embargo, aún no se conoce con precisión este potencial para utilizar dicha información en los procesos de toma de decisiones que contribuyan al desarrollo nacional de la bioenergía (Sánchez, 2021).

En el año 2021, se desarrollaron en Sancti Spíritus dos sistemas con el objetivo de conocer con precisión el potencial implementable del biogás en estos dos sectores y en el urbano. Dichos sistemas tienen funcionalidades en común, una interfaz similar y existe entre los dos una estrecha relación.

La migración de software describe el cambio de un software antiguo a una nueva aplicación. Un sistema de software existente se transfiere parcial o completamente a un nuevo entorno de destino. El trasfondo de una migración de software es que el sistema anterior debe ser

reemplazado por el nuevo, por ejemplo porque el sistema heredado está desactualizado, porque el nuevo entorno ofrece una funcionalidad adicional o mejor o porque los requisitos del sistema en general han cambiado tanto que la migración es la solución más eficiente (García, 2021).

A estos sistemas heredados se les conoce como aplicaciones informáticas que tienen su período de funcionamiento dentro de las organizaciones empresariales. Estas aplicaciones abarcan una tendencia de desarrollo en conjunto con la empresa y, con el paso del tiempo, a pesar de su desarrollo, la aplicación llega al final, donde la organización toma la decisión de mantenerse o evolucionar de plataforma mediante una migración dirigida a nuevas tecnologías (Tarazona, 2017).

Actualmente, las migraciones de datos se inician a menudo a medida que las empresas pasan de la infraestructura y las aplicaciones on-premises, al almacenamiento y aplicaciones basados en la nube para optimizar o transformar su empresa (Pumasunta Chungandro, 2020).

En el 2021, como se mencionó anteriormente, fueron desarrollados dos aplicaciones que facilitan el cálculo y la geolocalización de las entidades que generan residuos para la obtención del biogás. Sin embargo, estas aplicaciones presentan complicaciones en el sistema de información geográfico usado, que, al no tener la información muy exacta, genera complicaciones a la hora de ubicar las entidades que generan los residuos, y no muestra la información del potencial en el mapa. Al parecer, de la manera en que fueron implementados los sistemas, provocó lentitud en la interacción con las interfaces del usuario.

Al estar estos sistemas separados, de implementarse en el CEEPI se va a dificultar su mantenimiento, además de ocupar mucho espacio en memoria, pues se estaría instalando dos veces sistemas similares, pero solamente con diferentes datos y, se hace engorroso ya que tienen la misma finalidad: el cálculo del biogás, incluso interfaces similares, por lo que se dificulta el acceso a la información de los sistemas. Existen también complicaciones en el diseño de la base de datos, por lo que esto provoca la inconsistencia de la información almacenada. También en estos sistemas no se desarrolló un reporte general que, dentro de sus funcionalidades, exporte los datos a un documento que se pueda filtrar los datos para ser usado luego, por ejemplo, un Excel.

Esta tesis es la continuación de un proceso que tiene como finalidad facilitarle el trabajo a las personas que calculan el potencial de biogás, y, por tanto, debe también acortar el tiempo en horas de trabajo que se emplea en el uso de sistema de cómputo y debe mejorar los sistemas

anteriores. Con esta finalidad se llega a la conclusión de que es necesaria una migración de estos softwares para hacer una actualización, donde se vinculen ambos en un solo sistema. Pero no solamente quedarnos en el ámbito de las tecnologías en las que fueron implementados, sino, que para desarrollar esta actualización se realizará un cambio de tecnologías.

Dada la problemática anterior se define el **siguiente problema de investigación**:

Limitaciones en los sistemas para el cálculo y geolocalización del potencial de biogás en el CEEPI.

Para dar solución al problema de investigación planteado, se define el siguiente **objetivo general**: Migrar los sistemas existentes en el CEEPI para el cálculo y geolocalización del potencial de biogás para integrarlos y mejorar sus limitaciones de funcionamiento.

Para dar cumplimiento al objetivo general se definen las siguientes **preguntas de investigación**:

1. ¿Cuáles son los fundamentos teóricos, metodológicos y tecnológicos que sustentan el la migración de sistemas que calculan el potencial del biogás y geolocalización de entidades?
2. ¿Cómo diseñar una aplicación web que migre los sistemas de cálculo de potencial de biogás y la geolocalización de entidades ?
3. ¿Cómo desarrollar una aplicación web que migre los sistemas de cálculo de potencial de biogás y la geolocalización de entidades?

En respuesta se muestran las siguientes **tareas de investigación**:

1. Determinar los fundamentos teóricos, metodológicos y tecnológicos que sustentan la migración de sistemas que calculan el potencial del biogás y la geolocalización de entidades
2. Diseñar una aplicación web que migre los sistemas de cálculo de potencial de biogás y la geolocalización de entidades
3. Desarrollar una aplicación web que migre los sistemas de cálculo de potencial de biogás y la geolocalización de estas entidades

Este informe de investigación está compuesto por tres capítulos. El Capítulo 1 enuncia los fundamentos para la migración de los sistemas informáticos donde se encuentran todos los conceptos básicos a saber para llevar a cabo la migración. El Capítulo 2 trata sobre la planifi-

cación de la migración y todos los requisitos a tener en cuenta para la planificación y el diseño del nuevo software. En el Capítulo 3 se muestra la planificación de las pruebas y la implementación del software. El informe finaliza ofreciendo unas conclusiones generales y se plantean recomendaciones para el trabajo futuro.

CAPÍTULO 1

Fundamentos para la migración de dos sistemas informáticos

En el desarrollo de este capítulo se presentan las bases teóricas que sustentan la investigación. Se abordan aspectos tecnológicos y científicos que permiten enfocar la atención en los conocimientos necesarios que se tuvieron en cuenta. Se realiza un análisis del estado del negocio, tomándose en consideración diferentes herramientas y sistemas existentes con una finalidad similar, para proceder a su estudio y analizar posibles mecanismos de implementación. Se describe la metodología de desarrollo de software, así como las herramientas y tecnologías que serán empleadas durante el desarrollo de la investigación.

1.1. Biogás, producción y utilización

Cuando a finales del siglo XVIII el físico italiano Alessandro Volta identificó por primera vez el metano (CH_4) como el gas inflamable en las burbujas que emergían de los pantanos, no se pudo imaginar la importancia que este gas podría llegar a tener para la sociedad humana en los siglos venideros. El metano alcanzó una especial importancia durante la segunda guerra mundial debido a la escasez de combustibles. Con el fin de la guerra y la fácil disponibilidad de combustibles fósiles, la mayoría de las instalaciones fueron cesando en su funcionamiento. Sin embargo, en India, a comienzos de la década de los 60, se impulsó notablemente la tecnología

1.1 Biogás, producción y utilización

de producción de biogás a partir de estiércol bovino con el doble propósito del aprovechamiento energético y la obtención de un biofertilizante. En China, a inicios de la década de los 70, se ha fomentado la construcción de digestores, mediante programas de ámbito nacional (Varnero Moreno, 2012).

En los países industrializados la historia de la tecnología de biodigestión ha sido diferente y el desarrollo ha respondido más bien a motivaciones medioambientales que puramente energéticas, constituyendo un método clásico de estabilización de lodos activos de las plantas de tratamiento de aguas residuales domiciliarias. Durante la década de los ochenta, volvió a adquirir cierta importancia como forma de recuperación energética en explotaciones agropecuarias y agroindustriales. Sin embargo, con la disminución de los precios del petróleo, a finales de los años ochenta, el interés por la tecnología de digestión anaeróbica volvió a decaer, aunque en algunos países industrializados se han desarrollado importantes programas de desarrollo de plantas anaeróbicas a escala industrial y doméstica. En la actualidad, el biogás se utiliza en todo el mundo como una fuente de combustible tanto a nivel industrial como doméstico. Su explotación ha contribuido a impulsar el desarrollo económico sostenido y ha proporcionado una fuente energética renovable alternativa al carbón y el petróleo (Hurtado, 2013).

La actividad agropecuaria y el manejo adecuado de residuos rurales pueden contribuir significativamente a la producción y conversión de residuos animales y vegetales (biomasa) en distintas formas de energía. Durante la digestión anaeróbica de la biomasa, mediante una serie de reacciones bioquímicas, se genera el biogás, el cual, está constituido principalmente por metano (CH_4) y dióxido de carbono (CO_2). Este biogás puede ser capturado y usado como combustible y/o electricidad. De esta forma, la digestión anaeróbica, como método de tratamiento de residuos, permite disminuir la cantidad de materia orgánica contaminante, estabilizándola (bioabonos) y al mismo tiempo, producir energía gaseosa (biogás). Desde una perspectiva de los países desarrollados y en desarrollo, la biotecnología anaeróbica contribuye a cumplir tres necesidades básicas: a) Mejorar las condiciones sanitarias mediante el control de la contaminación; b) generación de energías renovables para actividades domésticas; y c) suministrar materiales estabilizados (bioabono) como un biofertilizante para los cultivos. Por lo tanto, la biotecnología anaeróbica juega un importante papel en el control de la contaminación y para la obtención de valiosos recursos: energía y productos con valor agregado (Varnero Moreno, 2012).

Digestión anaeróbica

La digestión aeróbica consiste en procesos realizados por diversos grupos de microorganismos, principalmente bacterias y protozoos que, en presencia de oxígeno actúan sobre la materia orgánica disuelta, transformándola en productos finales inocuos y materia celular. Al comienzo, el proceso de digestión aeróbica tuvo escasa aceptación, debido a que se desconocían sus principios fundamentales, además de que encarecían los costos del tratamiento por la cantidad adicional de energía necesaria para el suministro de aire al proceso. En contraste, los procesos de digestión anaeróbica permiten utilizar el metano generado como fuente de energía. La principal ventaja del proceso aeróbico es la simplificación en las operaciones de disposición de los lodos comparada con la relativa complejidad operativa del proceso de digestión anaeróbica (Skanfei y Angelidaki, 2022) .

La digestión aeróbica es un proceso mediante el cual los lodos son sometidos a una aireación prolongada en un tanque separado y descubierto. El proceso involucra la oxidación directa de la materia orgánica biodegradable y la autooxidación de la materia celular. En las primeras fases del proceso de digestión aeróbica, cuando una población de microorganismos se pone en contacto con una fuente ilimitada de sustrato, los microorganismos se reproducen con una tasa de crecimiento poblacional logarítmico que sólo está limitada por su propia habilidad de reproducirse. La tasa de consumo de oxígeno aumenta rápidamente debido a la absorción y asimilación de materia orgánica para la síntesis de nueva masa protoplasmática. A medida que progresa la oxidación de la materia orgánica disponible, la tasa de crecimiento bacteriano empieza a disminuir. Las fuentes de carbono orgánico disponibles se hacen limitantes, y por consiguiente, también se presenta una disminución en la tasa de consumo de oxígeno. Cuando la cantidad de materia orgánica disponible es apenas suficiente para garantizar la subsistencia de las distintas especies de microorganismos, éstos comienzan a autooxidarse mediante su metabolismo endógeno (de Compostaje, 2017).

1.1.1. Cálculo del potencial implementable del biogás

Antecedentes

Las primeras menciones sobre biogás se remontan al año 1600 identificados por varios científicos como un gas proveniente de la descomposición de la materia orgánica. En el año 1890 se construye el primer biodigestor a escala real en la India y ya en 1896 en Exeter, Inglaterra, las lámparas de alumbrados públicos eran alimentadas por el gas recolectado de los digestores que fermentaban los lodos cloacales de la ciudad. Tras las guerras mundiales comienzan a difundirse en Europa las llamadas fábricas productoras de biogás cuyo producto se empleaba en tractores y automóviles de la época. En todo el mundo se difunden los denominados tanques Imhoff para el tratamiento de aguas cloacales colectivas. El gas producido se utilizó para el funcionamiento de las propias plantas, en vehículos municipales y en algunas ciudades se lo llegó a inyectar en la red de gas comunal Cálculo del potencial implementable del biogás (Hurtado, 2013).

En Cuba se aprecia un significativo potencial para la producción y utilización del biogás, basado en la diversidad y el volumen de residuos contaminantes que generan los sectores agropecuario y agroalimentario; sin embargo, aún no se conoce con precisión este potencial para utilizar dicha información en los procesos de toma de decisiones que contribuyan en el desarrollo nacional de la bioenergía. Predecir el potencial del biogás es un tema clave para la elaboración de programas de desarrollo de la tecnología, como primera etapa de evaluación de su viabilidad. La cantidad de biogás producido y el contenido de CH₄ (componente que le confiere las propiedades combustibles al biogás), en la fase gaseosa, dependen del residuo que se biodegrada y su estado de oxidación. A mejor biodegradabilidad y más bajo estado de oxidación, se producirá más metano. Es común encontrar en la literatura diversas definiciones sobre el concepto de potencial energético de la biomasa. Los potenciales se clasifican según la probabilidad de su utilización: El potencial teórico está determinado por la cantidad total de biomasa que es producida o generada en forma de residuos sin considerar restricciones técnicas o económicas. El potencial disponible corresponde a la cantidad de biogás que es posible producir a partir de la biomasa efectivamente disponible, aplicando restricciones a la capacidad de recolección, uso alternativo, etc. El potencial técnico corresponde a la energía eléctrica y térmica que es posible de generarse a partir del biogás producido, aplicando restricciones tecnológicas a la conversión, fundamentalmente en términos de eficiencias eléctrica y térmi-

ca de los equipos de cogeneración. El potencial económico está determinado por la cantidad de energía que puede ser producida en condiciones de rentabilidad económica. El potencial implementable es aquel que, siendo económicamente factible, es posible ser implementado, aplicando restricciones respecto del marco legal vigente, logísticas y de otro tipo (Sánchez, 2021; Suárez-Hernández, 2020).

Caracterización del sector agropecuario, agroindustrial y residuos sólidos urbanos

El sector agropecuario cuenta con varias fuentes de residuo (ganado mayor, ganado menor, porcino, avícola, cunícula). Y otra fuente de residuo fuera del sector agropecuario lo constituyen los residuos sólidos urbanos. El sector agroindustrial cuenta con varias industrias (granos, pesquera, azucarera, conserva, láctea, cárnica, bebidas y licores). De las cuales se adquieren residuos orgánicos y con este se calcula el potencial disponible de biogás a través de fórmulas. A partir de todos estos residuos, se calcula el potencial disponible de biogás mediante fórmulas. Para la realización de dicho cálculo se parte de una fórmula general :

$$P_{\text{biogás}} = \beta * M_{\text{residuo}} * Y_{\text{biogás}}$$

donde:

P biogás: potencial disponible de biogás expresado en m^3 por día (m^3/d)

β biogás: factor que expresa la cantidad de residuo que puede aprovecharse con respecto al total, expresado en % y dividido entre 100.

M residuo: cantidad de residuo generado, expresado en toneladas por día Y biogás: rendimiento teórico del biogás expresado en m^3 por tonelada de materia fresca (MD, 2015).

Con el potencial disponible de biogás calculado se obtiene el potencial implementable de biogás en el que a partir de los resultados el especialista decide la forma de empleo (equipos de calor, cocción de alimentos y generación de electricidad)

1.2. Geolocalización

La geolocalización básicamente consiste en conocer la posición geográfica o ubicación (o coordenadas) de algún objeto: sea un teléfono, Tablet, computadora portátil, auto a través de coordenadas que generalmente vienen de satélites, aunque también se pueden obtener por otros medios. Hoy en día, la mayoría de las personas poseen al menos un dispositivo que proporciona esta información: el teléfono móvil; ya que cualquier tipo de teléfono móvil, sea o no un smartphone, puede ser localizado, ya que existen varias maneras de ser geolocalizados (Sánchez, 2021).

Para comprender la idea de geolocalización, por lo tanto, primero hay que saber qué es un sistema de coordenadas geográficas. Así se denomina al sistema que permite referenciar un punto en la superficie del planeta Tierra a partir de dos coordenadas angulares: longitud (oeste u este) y latitud (sur o norte). De este modo, se establecen los ángulos laterales de la superficie respecto al centro del planeta y según su eje de rotación. La geolocalización, en definitiva, apela a las coordenadas: líneas que permiten la determinación de la posición de un punto en el espacio. Para ubicar el punto en cuestión, suele recurrirse a un hardware y un software que analiza, registra y almacena la información geográfica que se referencia (Beltrán López, 2016).

Habitualmente, la geolocalización marca la ubicación de un cuerpo en el espacio midiendo coordenadas de latitud, longitud y altura. Por lo general la noción se asocia al Sistema de Posicionamiento Global (conocido como GPS por sus siglas en inglés), que utiliza satélites para obtener las coordenadas. El GPS usa una red de 24 satélites que orbitan sobre la Tierra con trayectorias sincronizadas para lograr la cobertura de toda la superficie. Para obtener una geolocalización, el aparato conectado al GPS recibe señales de al menos tres satélites, calculando el tiempo que demoran las señales en llegar al equipo. Así puede medir la distancia existente hasta el satélite a través de la trilateración inversa. Al conocer las coordenadas de los tres satélites, se establece la posición real del punto de medición (Pérez Porto y Merino, 2022).

1.3. Migración

La migración de datos es el proceso mediante el cual realizamos una transferencia de datos de unos sistemas de almacenamiento de datos a otros, de unos formatos de datos a otros o entre

diferentes sistemas informáticos.

Habitualmente, un proyecto de migración de datos se lleva a cabo para reemplazar o actualizar servidores o equipos de almacenamiento, para una consolidación de un sitio web, para llevar a cabo el mantenimiento de un servidor o para reubicar un centro de datos.

El éxito en un proyecto de migración de datos dependerá en gran medida del nivel de comprensión que se llegue a alcanzar acerca del proceso y sus implicaciones. Conocer los retos que implica una iniciativa de este tipo es el primer paso. Entre los más importantes se encuentran los siguientes:

Migración de almacenamiento: en un proyecto de este tipo no debería existir problema, siempre y cuando la aplicación sólo utilice interfaces generales para acceder a los datos. En la mayoría de los sistemas esto no es ningún inconveniente, aunque, cuando se trate de aplicaciones antiguas que se ejecutan en sistemas propietarios, sí podría serlo.

En ese caso, la iniciativa se complicaría y sería preciso llevar a cabo las pruebas necesarias antes de liberar la solución en producción. Principalmente por dos motivos:

- El código fuente de la aplicación puede no estar disponible.
- Podría darse el caso de que el proveedor de la aplicación no continuase activo en el mercado

Migración de la base de datos: se trata de una de las formas más sencillas de movimiento de datos, siempre y cuando la database se utilice como almacenamiento. Sin embargo, a pesar de la aparente simplicidad del proceso, pueden surgir contratiempos relacionados con:

- **Tipos de datos no coincidentes** (fecha, número, sub-registros): en este caso habría que trabajar por mantener la integridad de los datos, pudiendo ser preciso el orientar la gestión a la modificación de algunas de las aplicaciones que utilizan la base de datos.
- **Diferentes conjuntos de caracteres** (codificaciones distintas en cada columna para una misma tabla): cuando esto suceda habrá que revisar a fondo las aplicaciones que utilizan la base de datos.

Migración de aplicaciones: al enfrentarse a una iniciativa de este tipo es imperativo el recurrir a un proceso ETL completo. Esto es así debido a que, incluso cuando las aplicaciones están diseñadas por el mismo proveedor, almacenan datos en formatos y estructuras significativamente

diferentes. Esta particularidad complica la transferencia de datos. El paso de transformación, por ejemplo, es uno de los principales inconvenientes y, aunque apoyarse en una herramienta ETL otorga la ventaja de su conectividad, que la hace estar lista para usar con fuentes y destinos de datos dispares; las dificultades pueden aparecer al migrar datos de sistemas mainframe o aplicaciones usando determinadas formas de almacenamiento de datos, ya que:

- Los sistemas mainframe utilizan formatos basados en registros para almacenar datos que, aunque son sencillos de gestionar suelen incorporar optimizaciones.
- Las optimizaciones incluyen almacenamiento de números decimales codificados binarios, almacenamiento no estándar de valores de números positivos / negativos o almacenamiento de los subíndices mutuamente excluyentes dentro de un registro. Es decir, aspectos que complican la migración de datos (Power Data, 2018).

En el desarrollo de este proyecto se hará una migración de códigos heredados orientado las nuevas tecnologías de desarrollo de software, por lo que una migración de códigos es el movimiento del código de programación de un sistema a otro. Existen tres niveles distintos de migración de código con una complejidad, un costo y un riesgo crecientes:

1. La migración simple implica el paso del idioma a una versión más nueva.
2. Un segundo nivel de migración más complicado implica pasar a un lenguaje de programación diferente.
3. La migración a una plataforma o sistema operativo completamente nuevos es el tipo de migración más complejo.

El primer tipo de migración de código es un simple movimiento de una versión de un idioma a una versión más nueva, pero sintácticamente diferente. Esta es la ruta de migración más fácil ya que la estructura básica y muchas de las construcciones de programación generalmente no cambian. En muchos casos, el código antiguo realmente funcionaría, pero las rutinas nuevas y mejoradas o la modularización se pueden mejorar modificando el código para que se ajuste a la naturaleza del nuevo lenguaje. Por lo tanto, migrar el código conduciría a una ejecución más eficiente (Spiegato, 2022).

1.4. Sistema Heredado

Un sistema legacy o sistema heredado es un sistema, tecnología o aplicación de software antiguo o desactualizado que sigue en uso dentro de una organización porque sigue desempeñando las funciones para las que fue diseñado. Por lo general, los sistemas legacy ya no cuentan con soporte y mantenimiento y están limitados a nivel de crecimiento. Sin embargo, no pueden reemplazarse fácilmente.

Dado que los negocios evolucionan constantemente debido a cambios en la economía, nuevas leyes, el estado del mercado, la gestión, las reestructuraciones, etc., los sistemas tienden a quedar obsoletos con el paso del tiempo. A fin de adaptarse a todo estos cambios, los sistemas IT también están en continua evolución. Esto dificulta que los responsables IT tengan una visión completa del sistema porque las adaptaciones y actualizaciones suelen ser implementadas por diferentes personas a lo largo del tiempo (Menendez, 2017).

Los sistemas legacy suelen ser esenciales para el negocio. Esa es sin duda una de las principales razones por las cuales el uso de sistemas legacy sigue estando muy extendido entre las empresas. Por lo tanto, los responsables IT deben analizar cuáles son sus sistemas heredados y hasta qué punto merece la pena mantenerlos. En general, los sistemas legacy son críticos para las operaciones diarias, de modo que su migración o sustitución deben evaluarse y planearse con sumo cuidado para minimizar riesgos potenciales (Wagner, 2014).

La migración de datos es esencial a la hora de actualizar y migrar un sistema legacy.

Para garantizar que sea un éxito, es importante:

- Asegurarse de que todos los datos existentes se pueden extraer de forma segura.
- Transformar los datos para que sean compatibles con los nuevos formatos.
- Limpiar los datos para evitar posibles problemas de calidad (datos incompletos, duplicados o sin un formato adecuado).
- Importante un grupo de datos para comprobar que no hay errores y garantizar que la migración se desarrolla tal y como está planeado, antes de cargar todos los datos en el nuevo sistema.

1.4.1. Tipo de migraciones de software heredados

Según el «Estudio Comparativo de Metodologías de Desarrollo de Software con base en la Migración de Sistemas Heredados», existen las siguientes metodologías para la migración de sistemas heredados:

1. Migración de sistemas heredados fundamentada en el uso de herramientas de descubrimiento del conocimiento en bases de datos (KDD)
2. Migración de sistemas heredados MADIISH.

En la actualidad se han diseñado varias metodologías para la migración de un Sistema Heredado (SH), tal es el caso de la metodología de apoyo basada en el uso de las herramientas KDD (Knowledge Discovery in Databases) donde se señala que uno de los factores de éxito de un proyecto de migración es el entendimiento del SH, esto es, entender tanto el modelo de datos como el modelo de negocios que trata de cubrir el mismo. Mediante esta estrategia se pretende reconstruir algunos aspectos básicos del SH a migrar, de modo que sea posible entender el modelo de datos del SH, entender el modelo de negocios que intentaba cubrir el SH y determinar el nivel de calidad de los datos del SH. Esta metodología propone apoyar la recuperación de requisitos de un SH, basada en el uso de herramientas de minería de datos; dichas herramientas se basan en un esquema a lo cual se identifica la estructura y el objetivo de la realización de la minería de datos, la cual pretende “excavar” entre los datos para hallar información oculta y que posiblemente sea de gran utilidad en la toma de decisiones (Castillo Oporto y Figueredo Denis, 2020).

1.4.2. Metodología MADIISH para la migración de sistemas heredados

Para explicar el uso de esta metodología, primeramente se va a hacer un estudio comparativo de algunas metodologías muy usadas actualmente.

Metodologías Robustas

Se basan en una planificación con su respectiva documentación exhaustiva en cumplimiento de todo el proceso desde su inicio hasta obtener el resultado final del proyecto, entre algunas de estas metodologías robustas se encuentran:

- Microsoft Solution Framework (MSF)
- Métrica 3
- Proceso Unificado de Desarrollo (RUP)

Entre estas metodologías señaladas cuenta con ítems a seguir para su desarrollo:

1. Análisis de los requisitos: Consiste en documentar lo que el software deberá realizar al término de su migración.
2. Diseñar el sistema y programa: Es la realización de un prototipo y los algoritmos a utilizar sin codificar.
3. Codificar: Se realiza la escritura del código necesario para el desarrollo del software.
4. Ejecución de pruebas: Para valorar y localizar posibles errores o validaciones que no hayan sido consideradas. De igual forma permitirá conocer los tiempos de ejecución y la veracidad de los resultados obtenidos.
5. Verificación: Una vez que fue probado por el programador, se instalará para que el usuario realice pruebas reales.
6. Mantenimiento del sistema nuevo: Normalmente no se prueban todos los posibles casos por lo que siempre habrá que corregir errores y realizar actualizaciones.
7. Amplia documentación en todo momento.

Metodologías Ágiles

Se basan en una comunicación estrecha en los participantes y se orientan en el resultado del producto y no en la documentación, proponiendo cambios de último momento entre estas metodologías:

1. Extreme Programming (XP).
2. Scrum.
3. Crystal Metodología.

En cambio, las metodologías ágiles consisten principalmente en:

1. Trabajar para obtener software funcional en lugar de demasiada documentación.

2. Colaboración con el usuario para la comprensión rápida de sus procesos de negocios.
3. Se tiene la posibilidad de hacer cambios de planes en cualquier punto del proyecto evitando la planeación extensa, lo que permite iniciar la programación.

Sin embargo, aunque sus ventajas son interesantes, estas metodologías también presentan inconvenientes que hay que asumir cuando se decide trabajar con ellas. Estas son:

1. Falta de documentación del diseño
2. Problemas de comunicación debido a la participación del usuario donde se pueden dar malas interpretaciones en lo hablado y no documentado
3. Existe una fuerte dependencia de las personas, si el usuario no tiene el tiempo suficiente disponible puede alentar el desarrollo del proyecto
4. Falta de reusabilidad derivada de la falta de documentación

Estos inconvenientes son contemplados y existe un enfoque especial a los mismos dentro de la metodología de migración MADIISH.

En esta metodología por naturaleza tiene una forma incremental e iterativa en cada una de sus iteraciones en cada una de sus fases. En cuanto a procesos ágiles por su organización y entrega del trabajo continuamente, mediante una investigación del sistema heredado se puede contar con la obtención de información y definiciones de módulos que integran al sistema heredado. Mediante la fase de esta metodología se puede realizar un análisis de acuerdo a la selección de módulos principales en los que abarcan información categorizada, planteando requerimientos y mejoras comprendiendo funciones del negocio. La metodología MADIISH está basada en las principales características de las siguientes metodologías:

- Metodología iterativa debido a que se pueden realizar modificaciones constantes sobre un mismo subsistema hasta la entrega satisfactoria al usuario final.
- Metodología incremental ya que se migra subsistema por subsistema
- Metodología ágil por la participación constante del cliente para reducir la documentación.
- La metodología MADIISH presenta una estructura propia de migración a nuevas tecnologías.

Para el presente proyecto se procede a seleccionar la metodología de migración MADIISH que abarca en gran ámbito todos los procesos de trabajo a desarrollar contando con un plan guía

enfocado en etapas contempladas en la propuesta (Power Data, 2018).

1.5. Herramientas Informáticas

Node.JS

A pesar de que la primera característica del lenguaje php, -lenguaje que fue utilizado para el desarrollo de los sistemas-, es su rapidez; según el índice TIOBE en febrero de 2022, (indicador de la popularidad de los lenguajes de programación que se actualiza cada mes y sus ratings se basan en el número de ingenieros habilitados a nivel mundial, así como de cursos y proveedores), la posición de este lenguaje está en un 1,79%, que queda por debajo del lenguaje JavaScript con un 1,83%; lenguaje que se usará en el desarrollo del nuevo sistema.

Además, según el informe anual de GitHub de los lenguajes de programación más usados, JavaScript se mantiene en el número uno, mientras que php ha estado descendiendo en su uso desde 2019. Para complementar el uso de javascript se ha incorporado Node.JS que es un entorno de ejecución de un solo hilo, de código abierto y multiplataforma para desarrollar aplicaciones rápidas y escalables del lado del servidor y de la red, y que además está escrito en los lenguajes de programación C y C++ (14.08% y 8.01% en TIOBE) junto con JavaScript.

Pero no solamente es este índice nuestra fuente de información, sino que según las perspectivas tecnológicas de la agencia de trabajos a distancia DistantJob, el rendimiento de Node.JS es, la mayoría de las veces, mejor. Aunque el rendimiento también es en gran medida el resultado de las habilidades y la experiencia del equipo de desarrollo, Node.JS ofrece varias ventajas sobre PHP en términos de rendimiento que van más allá de la habilidad del desarrollador:

1. Menos dependencias,
2. Sin intérprete (Node.JS es más ágil y más pequeño. Este es un resultado de Google. El gigante tecnológico hizo una gran contribución al rendimiento del motor JavaScript V8.
3. Flujo de entrada/salida sin bloqueo y controlado por eventos. (Github, 2022; Ochoa, 2022)

Node.JS es un entorno de ejecución de un solo hilo, de código abierto y multiplataforma para desarrollar aplicaciones rápidas y escalables del lado del servidor y de la red. Consiste en una arquitectura de E/S (entradas y salidas) basada en eventos y sin bloqueos, y funciona con el

motor de ejecución de JavaScript V8 de Google. Como resultado, es una herramienta eficiente e ideal para el desarrollo de aplicaciones en tiempo real. Además, Node.JS está escrito en los lenguajes de programación C y C++ junto con JavaScript. Esto se debe a que el C de bajo nivel es uno de los lenguajes de programación más eficientes para construir sistemas operativos, motores de bases de datos y todos los demás códigos relevantes para la programación de sistemas. Por otra parte, C++ tiene varias características que permiten a Node.JS comunicarse con el sistema operativo para establecer una conexión interna directamente con el servidor, lo que es inconcebible para un lenguaje de programación del lado del cliente como JavaScript. Como resultado, interactúa con C++ para gestionar estas características informáticas y da lugar a Node.JS, un entorno de programación súper eficiente tanto del lado del servidor como del lado del cliente (Almeida, 2022).

Node.JS, Fundamentos básicos

Node.JS es de código abierto: esto significa que el código fuente de Node.JS está disponible públicamente. Y es mantenido por colaboradores de todo el mundo. La guía de contribución de Node.JS le muestra cómo contribuir.

Node.JS es multiplataforma: Node.JS no depende de ningún software de sistema operativo. Puede funcionar en Linux, macOS o Windows.

Node.js es un entorno de tiempo de ejecución de JavaScript: cuando escribe código JavaScript en su editor de texto, ese código no puede realizar ninguna tarea a menos que lo ejecute (o ejecute). Y para ejecutar su código, necesita un entorno de tiempo de ejecución.

Los navegadores como Chrome y Firefox tienen entornos de tiempo de ejecución. Es por eso que pueden ejecutar código JavaScript. Antes de que se creara Node.js, JavaScript solo podía ejecutarse en un navegador. Y se usó para construir solo aplicaciones front-end.

Node.js proporciona un entorno de tiempo de ejecución fuera del navegador. También se basa en el motor de JavaScript Chrome V8. Esto hace posible crear aplicaciones de back-end utilizando el mismo lenguaje de programación JavaScript con el que puede estar familiarizado (Semah, 2022).

Vue.JS

En el uso de Node.JS se desarrollará un frontend, usando Vue.JS (framework progresivo, que puede ser empleado en pequeños, o grandes proyectos, no define de manera estricta una estructura de directorios, ni viene con componentes o paquetes por defecto, sino que permite que los desarrolladores vayan integrando los componentes que requieren de acuerdo a sus necesidades) con Nuxt.JS (framework para crear aplicaciones Vue.js de una manera más rápida, fácil y organizada); en conjunto con Vuetify.JS (biblioteca de Vue.JS destinada a la creación de interfaces de usuario que implementa componentes de usabilidad listos para usar y combina las bondades de Vue.JS y Material Design). Además, Vuetify es un marco de componentes para Vue.JS 2. Su objetivo es proporcionar componentes limpios, semánticos y reutilizables que facilitan la creación de su aplicación. Vuetify utiliza el patrón de diseño Material Design de Google, tomando como referencia otros marcos populares como Materialize.css, Material Design Lite y Semantic UI. Además se incorpora el uso de Tailwind.css para el tema de nuestra aplicación como una dependencia más del software.

A pesar de que el framework de desarrollo usado para estos sistemas (Symfony) tiene su última actualización en mayo de 2022, siendo ya la versión 6.1.7.; y que este framework permite el uso de API REST (interfaz que dos sistemas de computación utilizan para intercambiar información de manera segura a través de Internet), para la realización de la API REST se usará Strapi.io para el backend, que es un CMS(Sistema de Gestión de Contenido) sin encabezado de código abierto basado en Node.js y usado para diseñar APIs de manera rápida y sin una sola línea de código.

Nuxt.JS

Nuxt.JS es el framework Vue más intuitivo disponible en la actualidad. Combina la potencia de Vue.JS con las características de renderizado del lado del servidor para hacerlo más potente. Se puede construir una aplicación completa de renderizado del lado del cliente de Vue.js, una aplicación completa generada estáticamente y una aplicación monolítica.

Nuxt.JS resuelve el problema de la estructuración de un proyecto Vue.JS, ya que viene con una arquitectura de desarrollo frontend preparada para la empresa. Las características de Nuxt.js ya están estructuradas utilizando los estándares de la industria para crear aplicaciones empre-

sariales.

Nuxt.JS funciona de la misma manera que un framework del lado del servidor cuando un usuario visita un sitio web. Si la renderización del lado del servidor está activada, las peticiones se renderizan en el servidor cada vez que el usuario solicita una página, por lo que se necesita un servidor para poder servir la página en cada petición. Además, si se habilita la renderización del lado del cliente, se renderiza el contenido de la página en el navegador utilizando JavaScript.

Strapi.io

Strapi es un CMS sin encabezado de código abierto basado en node.js para todas las necesidades de API y gestión de contenidos de los desarrolladores. Permite crear API funcionales en cuestión de minutos y desarrollar software sin la molestia de las complejidades de un CMS. Strapi es para empresas de todos los tamaños, desde trabajadores autónomos hasta grandes organizaciones que necesiten administrar contenido a través de una API. Strapi se basa en los comentarios de la comunidad, desarrolladores y gestores de contenido.

Strapi está diseñado para empresas y proyectos de todos los tamaños, desde startups a grandes organizaciones que busquen manejar su contenido a través de una API. Ha sido utilizado en el desarrollo de proyectos en curso y otros como UEP Mallorca, App de Eventos a nivel local. Es uno de los *Headless CMS* más populares en la actualidad. Gestor de contenidos independiente de la capa de presentación. Nos permite trasladar el código a través de múltiples servidores sin perder información, y reutilizar componentes con una exportación sencilla para cualquier futuro cambio de CMS (Russi, 2022).

Concepto de API REST

Una API de REST, o API de RESTful, es una interfaz de programación de aplicaciones (API o API web) que se ajusta a los límites de la arquitectura REST y permite la interacción con los servicios web de RESTful. El informático Roy Fielding es el creador de la transferencia de estado representacional (REST).

Las API son conjuntos de definiciones y protocolos que se utilizan para diseñar e integrar el software de las aplicaciones. Suele considerarse como el contrato entre el proveedor de información y el usuario, donde se establece el contenido que se necesita por parte del consumidor

(la llamada) y el que requiere el productor (la respuesta). Por ejemplo, el diseño de una API de servicio meteorológico podría requerir que el usuario escribiera un código postal y que el productor diera una respuesta en dos partes: la primera sería la temperatura máxima y la segunda, la mínima.

En otras palabras, las API le permiten interactuar con una computadora o un sistema para obtener datos o ejecutar una función, de manera que el sistema comprenda la solicitud y la cumpla (Red, 2020).

Vuetify.JS

Vuetify es un framework que combina la potencia del popular VueJs con la estética de Material Design. Permite acelerar el desarrollo de aplicaciones web complejas, incorporando una gran cantidad de componentes “listos para usar”.

Material Design es una norma para el diseño enfocado en la visualización en móviles con el sistema operativo Android, además de su implementación en plataformas web. Esta norma fue implementada por Google en junio de 2014. La unión de estas dos tecnologías, permiten acelerar el desarrollo de aplicaciones web complejas, incorporando una gran cantidad de componentes.

Vuetify ordena el layout de nuestra página mediante el sistema grid, el cual se ajusta a una gran cantidad de resoluciones de pantalla. Dispone de una enorme librería de componentes que incluye elementos desde botones, sliders, diferentes tipos de inputs, a componentes más avanzados como card, que nos permitirán habilitar toda una serie de componentes para interactuar con los usuarios (Llamas, 2019).

1.6. Conclusiones parciales

En el desarrollo de este capítulo se definieron los conceptos básicos a tratar en el desarrollo de la investigación, así como la metodología de migración de sistemas heredados a utilizar. Además quedaron definidas las herramientas y tecnologías informáticas a utilizar en la implementación del nuevo sistema migrado.

CAPÍTULO 2

Planificación, diseño e implementación de la nueva aplicación web

En este capítulo se planifica cómo va a quedar el nuevo sistema migrado, a partir del uso de herramientas para la migración como es la metodología MADIISH. Se explican la lógica del negocio, los diferentes tipos de factibilidades: técnica, operacional, económica financiera; se explican los requerimientos del nuevo software, en funcionales y no funcionales; y además se va entrando en lectura de las fases de la metodología MADIISH que se van a ir aplicando en este capítulo que trata de la planificación y el diseño del nuevo sistema.

2.1. Factores a tener en cuenta para el desarrollo de la migración

Causas de la migración

Se definen causas de la migración, como son:

1. La no existencia de soporte técnico para la herramienta de desarrollo de algunas aplicaciones actuales en producción
2. Normativas vigentes

2.1 Factores a tener en cuenta para el desarrollo de la migración

3. Costos de mantenimiento

4. Imposibilidad de crecimiento y estabilidad tecnológica.

En el caso de la presente investigación el sistema se considera heredado debido a que actualmente las tecnologías que maneja el sistema están en menor grado obsoletas en el contexto

Lógica del negocio

Existen dos sistemas ya implementados en Symphony que calculan el potencial del biogás, es necesario migrar estos sistemas y añadir nuevas funcionalidades tanto para los reportes como para el manejo de los mapas. La información mostrada en los mapas es necesario mejorarla y añadir nuevos residuos que no están incluidos y que se calculan de una manera diferente.

Ventajas de la migración del sistema: Contenido más centralizado y fácil de usar y de visualizar

Factibilidad técnica

El centro de estudios de la energía y procesos industriales cuenta con la tecnología necesaria para usar el nuevo sistema, porque no se necesita demasiado avance tecnológico para implementarlo:

- Windows 10 64 bits
- Navegador Chrome o Firefox Actualizado(s)
- Velocidad normal de conexión normal para el software en general (si se desea usar la geolocalización se necesita una velocidad de más de 100 m/bits)

Factibilidad operacional

En el CEEPI se cuenta con los recursos humanos capacitados para usar el nuevo sistema, aunque también de tener alguna duda pueden hacer las preguntas necesarias

Factibilidad económica financiera

Al implementar el nuevo sistema en la empresa, será más rápido el cálculo del biogás y demás operaciones con las que contaban los sistemas anteriores, y que están presentes en este nuevo sistema.

Especificaciones de requerimientos

Propósito

2.1 Factores a tener en cuenta para el desarrollo de la migración

Al migrar estos sistemas y unificar su contenido, se podrán usar nuevamente pero de manera centralizada, sin complicaciones a la hora de acceder a un residuo o a entidades de residuos diferentes.

Ámbito del Software

Migración de sistemas de cálculo del potencial del biogás de Symfony v4 a Node js v16.14.2 + Integración de APIs y frontend independiente para el Centro de Estudios de la Energía y Procesos Industriales (CEEPI).

Nuevo en el sistema:

1. Software basado en el uso de API REST
2. Optimización y cambio de lenguaje de funciones de los sistemas anteriores
3. Mejoramiento de la estructura anterior del software
4. Exportar información de las entidades a excel además de poder filtrar antes de exportar
5. Mayor precisión en el sistema de información geográfico y muestra de la información de los potenciales en el mapa.
6. Se incorpora una búsqueda rápida en los potenciales calculados implementables y disponibles en un tipo de residuo determinado

Reglas del Negocio

Reglas que se relacionan y se heredan de los sistemas anteriores:

1. Cada tipo de residuo posee un potencial teórico específico
2. Cada entidad que genera un tipo de residuo tiene una geolocalización y la información resumida en el mapa
3. Cada usuario puede acceder a su contenido y el administrador no puede acceder y vice-versa
4. No se puede repetir al añadir un usuario ni su nombre de usuario ni su email

2.1 Factores a tener en cuenta para el desarrollo de la migración

5. Sólo el administrador del software puede añadir y editar la información de los especialistas

Funciones del Producto

Justificación de la migración

Para hacer esta migración de sistema se tienen los siguientes motivos:

1. Existen altos costos y demasiado tiempo para tener implementados los dos softwares independientes al mismo tiempo en el centro.
2. No se puede hacer una actualización porque las tecnologías de los sistemas están poco usadas en el contexto actual

Procedimiento propuesto

Esta migración se llevará a cabo mediante el siguiente procedimiento:

1. Determinar la causa de la migración
2. Recoger datos que son importantes al migrar
3. Consultar a los usuarios de la empresa para preguntar las razones por las que es necesario migrar los sistemas, y, de haberse migrado, la manera en la que esto va a afectar a los usuarios.
4. Planificar el procedimiento y el momento de realizar la migración
5. Evaluación de resultados

Estrategia de migración

En esta migración de sistema es necesaria una metodología, las herramientas informáticas y los métodos de prueba y personalización. La metodología permite que exista un orden y una planificación a seguir, para que se dividan las responsabilidades, se controlen los avances, y se reúna lo necesario para llevar a cabo la migración. Las herramientas informáticas son muy importantes para realizar la migración, porque brindan apoyo para minar datos, acortar tiempo en análisis y en elementos relacionados dentro del modelo del sistema.

Características de los usuarios del nuevo sistema

Tabla 2.1: Características de los usuarios del nuevo sistema

Nombre de usuario	Tipo de usuario	Área funcional	Actividad
Administrador	Administrador del sistema	Administración	<ol style="list-style-type: none">1. Administrar el sistema2. Administrar usuarios
Especialista	Usuario del sistema	Control de datos	<ol style="list-style-type: none">1. Control de entidades que generan residuos: Agropecuarios, Industriales, Urbanos2. Control del potencial y el uso final del biogás

En la Tabla 2.1 se muestran las características que va a tener cada usuario del sistema

2.2. Requisitos

2.2.1. Requisitos funcionales

1. Toda la información del sistema será manejada a través de APIs
2. La interfaz debe mostrar la información que se envíe a través de la API

3. Requisitos iguales a los sistemas anteriores:

R_1. Gestionar entidades pesqueras

- 1.1. Insertar entidades pesqueras
- 1.2. Modificar entidades pesqueras
- 1.3. Eliminar entidades pesqueras
- 1.4. Buscar entidades pesqueras
- 1.5. Mostrar entidades pesqueras

R_2. Gestionar usuarios

- 2.1 Insertar usuario
- 2.2 Modificar usuario
- 2.3 Eliminar usuario
- 2.4 Buscar usuario

R_3. Gestionar Entidades Porcinas

- 3.2 Insertar entidad porcina
- 3.3 Modificar entidad porcina
- 3.4 Eliminar entidad porcina
- 3.5 Buscar entidad porcina

R_4. Gestionar Entidades Avícolas

- 4.1 Insertar entidades avícolas
- 4.2 Modificar entidades avícolas
- 4.3 Eliminar entidades avícolas
- 4.4 Buscar entidades avícolas

R_5. Gestionar Entidades de Ganado Mayor

- 5.1 Insertar entidades de Ganado Mayor
- 5.2 Modificar entidades de Ganado Mayor
- 5.3 Eliminar entidades de Ganado Mayor
- 5.4 Buscar entidades de Ganado Mayor

R_6. Gestionar Entidades de Ganado Menor

- 6.1 Insertar entidades de Ganado Menor
- 6.2 Modificar entidades de Ganado Menor
- 6.3 Eliminar entidades de Ganado Menor

6.4 Buscar entidades de Ganado Menor

R_7. Gestionar entidades cunículas

7.1. Insertar entidades cunículas

7.2. Modificar entidades cunículas

7.3. Eliminar entidades cunículas

7.4. Buscar entidades cunículas

R_8. Gestionar entidades de Residuos Sólidos Urbanos

8.1. Insertar entidades de Residuos Sólidos Urbanos

8.2. Modificar entidades de Residuos Sólidos Urbanos

8.3. Eliminar entidades de Residuos Sólidos Urbanos

8.4. Buscar entidades de Residuos Sólidos Urbanos

8.5. Mostrar entidades de Residuos Sólidos Urbanos

R_9. Gestionar entidades arroceras

9.1. Insertar entidades arroceras

9.2. Modificar entidades arroceras

9.3. Eliminar entidades arroceras

9.4. Buscar entidades arroceras

9.5. Mostrar entidades arroceras

R_10. Gestionar entidades de granos

10.1. Insertar entidades de granos

10.2. Modificar entidades de granos

10.3. Eliminar entidades de granos

10.4. Buscar entidades de granos

10.5. Mostrar entidades de granos

R_11. Gestionar entidades de cultivos varios

- 11.1. Insertar entidades de cultivos varios
- 11.2. Modificar entidades de cultivos varios
- 11.3. Eliminar entidades de cultivos varios
- 11.4. Buscar entidades de cultivos varios
- 11.5. Mostrar entidades de cultivos varios

R_12. Gestionar entidades cárnicas

- 12.1. Insertar entidades cárnicas
- 12.2. Modificar entidades cárnicas
- 12.3. Eliminar entidades cárnicas
- 12.4. Buscar entidades cárnicas
- 12.5. Mostrar entidades cárnicas

R_13. Gestionar entidades de conserva

- 13.1. Insertar entidades de conserva
- 13.2. Modificar entidades de conserva
- 13.3. Eliminar entidades de conserva
- 13.4. Buscar entidades de conserva
- 13.5. Mostrar entidades de conserva

R_14. Gestionar entidades lácteas

- 14.1. Insertar entidades lácteas
- 14.2. Modificar entidades lácteas
- 14.3. Eliminar entidades lácteas
- 14.4. Buscar entidades lácteas
- 14.5. Mostrar entidades lácteas

R_15. Gestionar entidades azucareras

- 15.1. Insertar entidades azucareras
- 15.2. Modificar entidades azucareras
- 15.3. Eliminar entidades azucareras
- 15.4. Buscar entidades azucareras
- 15.5. Mostrar entidades azucareras

R_16. Gestionar entidades de bebidas y licores

- 16.1. Insertar entidades de bebidas y licores
- 16.2. Modificar entidades de bebidas y licores
- 16.3. Eliminar entidades de bebidas y licores
- 16.4. Buscar entidades de bebidas y licores
- 16.5. Mostrar entidades de bebidas y licores

R_17. Calcular potencial implementable

- 17.1. Equipos de Calor
- 17.2. Cocción de alimentos
- 17.3. Generación de electricidad mínima
- 17.4. Generación de electricidad máxima

R_18. Calcular potencial disponible

- 18.1. Mínimo
- 18.2. Máximo
- 18.3. Promedio

R_19. Iniciar sesión

R_20. Cerrar sesión

2.2.2. Requisitos no funcionales

Eficiencia:

- El sistema trabajará en dos servidores distintos, uno de la lógica y otro de la interfaz aumentando la velocidad de respuesta mediante la comunicación de APIs.
- El Sistema al contar con tecnología actualizada aumentará mucho su velocidad de respuesta.

Usabilidad:

- El sistema contará con la nueva interfaz, manteniendo algunas características de los sistemas anteriores

Seguridad:

- Solo personal asociado a la institución tendrá acceso al sistema y al manejo de la información.
- Solo el administrador e informático podrán eliminar información sensible en el sistema.

Mantenibilidad:

- El sistema estará montado en la versión más reciente de Node.JS v16.14.2
- Se le podrán agregar módulos y funciones que se necesiten, de acuerdo con el contexto.

2.2.3. Requisitos de Software

1. Lenguaje a utilizar: Javascript con Node.JS como entorno usando Vue.JS para frontend y Nuxt.JS como framework de Vue.JS y Vuetify.JS como framework de vuejs
2. Se usará postgresql como gestor de base de datos y Strapi para crear una interfaz y un servidor para el servicio API
3. Recursos de iconos : Material Design Icons (mdi-icons)
4. El sistema no permitirá usuarios que no sean previamente registrados dentro del sistema con la supervisión del administrador

2.3. Buenas prácticas para el desarrollo del nuevo software

En las buenas prácticas para realizar este nuevo software se incluye el cambio en los patrones de diseño y arquitectura para un mejoramiento de la interfaz del software y de la interacción con las interfaces de usuario.

2.4. Fases del ciclo de migración para el diseño del sistema

2.4.1. Fase Cero

Se define de forma abstracta la conformación del supra sistema en sus diferentes sistemas, así como la de sus sistemas en sub sistemas, lo cual permite establecer la relación entre cada sub sistema para facilitar la determinación de la jerarquía de migración y la interoperabilidad de dicho subsistema o sistema con otros.

Esta fase se denomina fase cero debido a que durante la migración se realiza una vez, al inicio de la migración. A su vez, permite el reconocimiento de las necesidades del cliente y se plantean las estrategias necesarias para poder llevar a cabo el proceso de migración (Pumasunta Chungandro, 2020).

La tabla 2.2 muestra las actividades a realizar en la fase cero de MAADISH.

Descomponiendo las funcionalidades del supra sistema en diferentes sistemas:

El nuevo sistema de cálculo del potencial del biogás en diferentes entidades, tiene integrados el sistema número uno, que incluye las entidades Agropecuarias y RSU (residuos sólidos urbanos), y el sistema número dos, que incluye las entidades Agroindustriales. Estos sistemas tienen funcionalidades en común, que son el cálculo del potencial disponible, la geolocalización de entidades y el cálculo del potencial implementable. Se incluye en este nuevo sistema la funcionalidad particular del sistema uno, que es el Reporte General, donde se podrá filtrar la información de las entidades y exportar a un Excel la información deseada. Dentro de cada sistema se encuentran sus entidades particulares pertenecientes a un tipo de residuo específico. En el nuevo sistema va a existir además la posibilidad de hacer un cambio de residuo en

2.4 Fases del ciclo de migración para el diseño del sistema

Tabla 2.2: Actividades que se realizan en la fase cero de MAADISH

Fase	Actividades
Recolección de Información	Esta fase brindará los datos necesarios para comenzar el proceso de migración: <ol style="list-style-type: none">1. Inventario del software2. Definición del orden de migración por precedencia de aplicaciones3. Detección del usuario final responsable del sistema4. Determinación del responsable técnico de cada sistema
Prioridad de migración	Se establece una escala de prioridad: 0: No es urgente su migración 1: Baja 2: Media 3: Alta

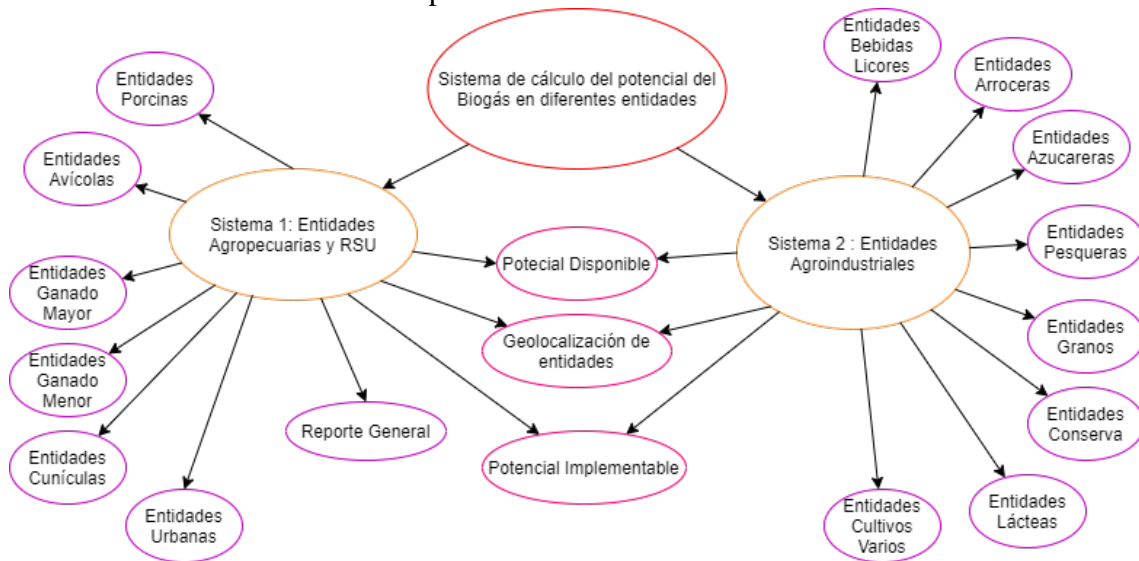
el proceso de registro de estas entidades al sistema, lo que provoca un cálculo diferente del biogás.

La figura 2.3 muestra la descomposición del sistema en sus diferentes subsistemas

El orden de migración está definido por el orden de las fases MAADISH, pero en la migración de estos sistemas como primera prioridad se realizó el arreglo y la unión de las bases de datos de los sistemas, luego se procedió a hacer un inventario de los requisitos funcionales del software, comenzando a crear los gestionar de las entidades del sistema más extenso y luego del más sencillo, luego se buscaron las fórmulas del potencial localizadas en el código de los sistemas y se aplicaron en los cálculos, dejando para último el reporte general en el rol del usuario del sistema, además de la geolocalización. Para el rol de administrador, se hizo la investigación de los municipios y provincias del país, para integrarlos en la base de datos, así como de la información que debe tener un usuario del sistema. Al terminar el software se aplicaron todas las funcionalidades de inicio y cierre de sesión, y las rutas protegidas del software.

2.4 Fases del ciclo de migración para el diseño del sistema

Tabla 2.3: Descomposición del sistema en sus diferentes sistemas



2.4.2. Fase 1: Análisis de procesos de negocio

Se elige un sistema a migrar tomando en cuenta los siguientes criterios: importancia, sencillez, conveniencia, complejidad, contingencia y selectividad

A su vez, se realiza un análisis de entradas y salidas de datos para determinar la dependencia de otros subsistemas. De igual forma se definen los nuevos requerimientos de usuario y se comprende mediante el análisis de procesos de negocio la operación del sistema, esquematizando el conocimiento sobre un diagrama de entradas y salidas de datos, permitiendo la eficiencia de los nuevos procesos (Pumasunta Chungandro, 2020).

En la tabla 2.4 se muestran las actividades que se realizan en la fase uno de MAADISH

En la tabla 2.1 se muestra el proceso de entrada y salida de un usuario al sistema según sus roles.

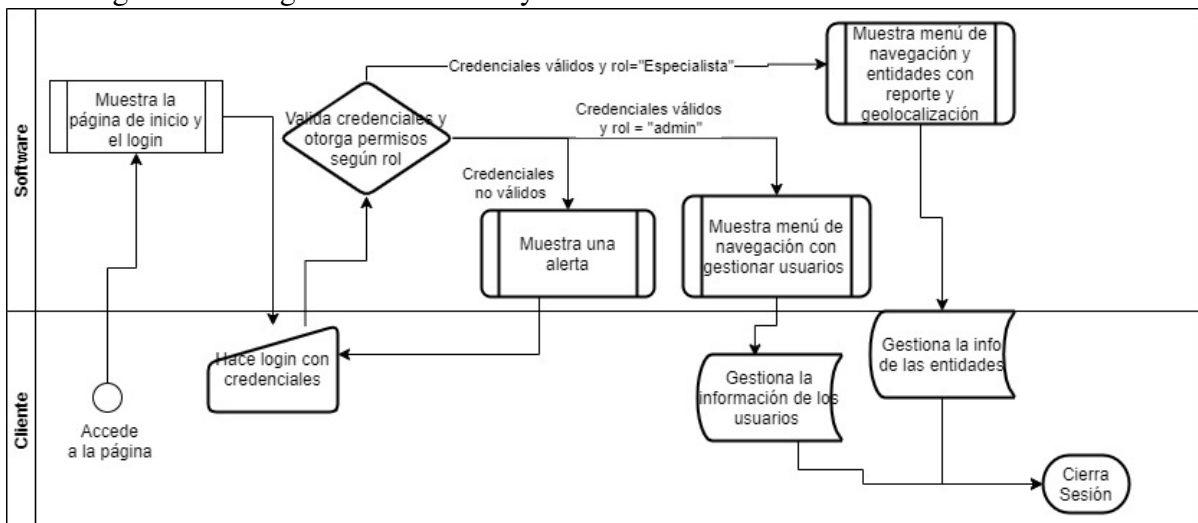
En la figura 2.2 se muestra un diagrama de entradas y salidas con el proceso de agregar una nueva entidad al sistema

2.4 Fases del ciclo de migración para el diseño del sistema

Tabla 2.4: Actividades que se realiza en la fase 1

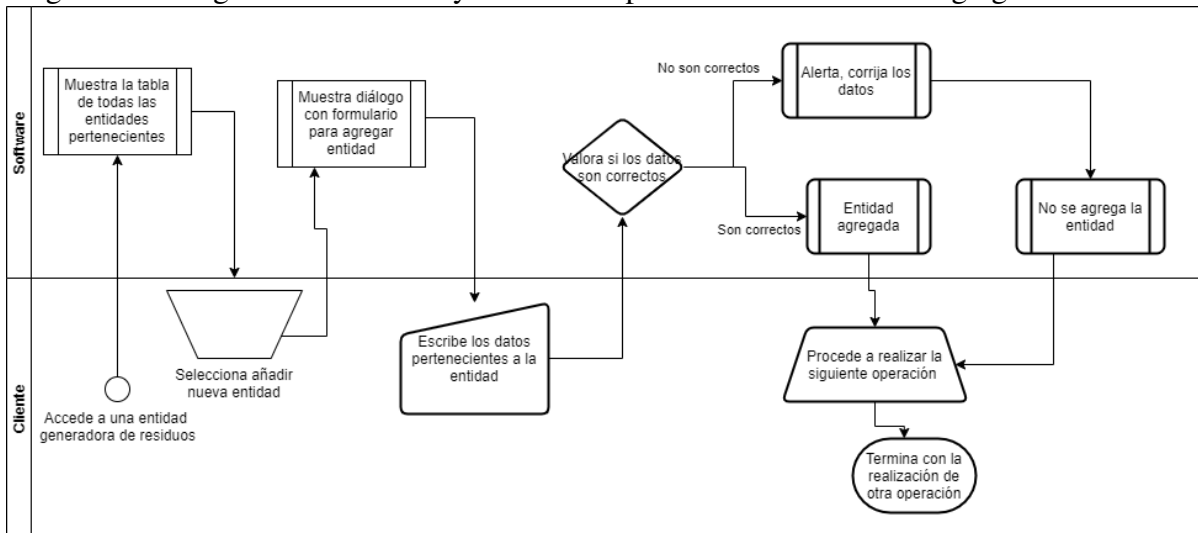
Fase	Actividades
Recolección de Información	Esta fase brindará los datos necesarios para comenzar el proceso de migración: <ol style="list-style-type: none"> 1. Categorización por Front End y Backend 2. Definición de la base de datos a ser usada 3. Definición del Frontend a ser usado
Capacitación	Capacitación del personal técnico involucrado por sistema

Figura 2.1: Diagrama de entradas y salidas del inicio de sesión en el sistema base



2.4 Fases del ciclo de migración para el diseño del sistema

Figura 2.2: Diagrama de entradas y salidas de operaciones del sistema. Agregar una entidad



Nuevos requerimientos del usuario:

Se mantienen todos los requerimientos funcionales de los softwares anteriores, añadiendo el filtro y el reporte general con el exportar a Excel. Se mantiene los derechos de ver información de cada usuario del sistema y del administrador.

2.4.3. Fase 2: Modelado, programación y generación del prototipo

Se realiza el modelado la base de datos o actualización de la misma en caso de existir una previa de algún subsistema migrado anteriormente, el desarrollo de interfaces de usuario, la programación para la comunicación entre interfaces con BDD y la generación de una versión prototipo para realizar pruebas y mostrar al usuario. Como primer modelo el prototipo esté dispuesto cambio de acuerdo a la necesidad de la organización, a su vez el subsistema migrado como nuevos módulos anexos podrán incluir registro de datos en versión pruebas para identificar el proceso (forma formulario web) cumpla sus requerimientos (Pumasunta Chungandro, 2020).

En la tabla 2.5 se muestran las actividades a realizar en la fase dos de MAADISH

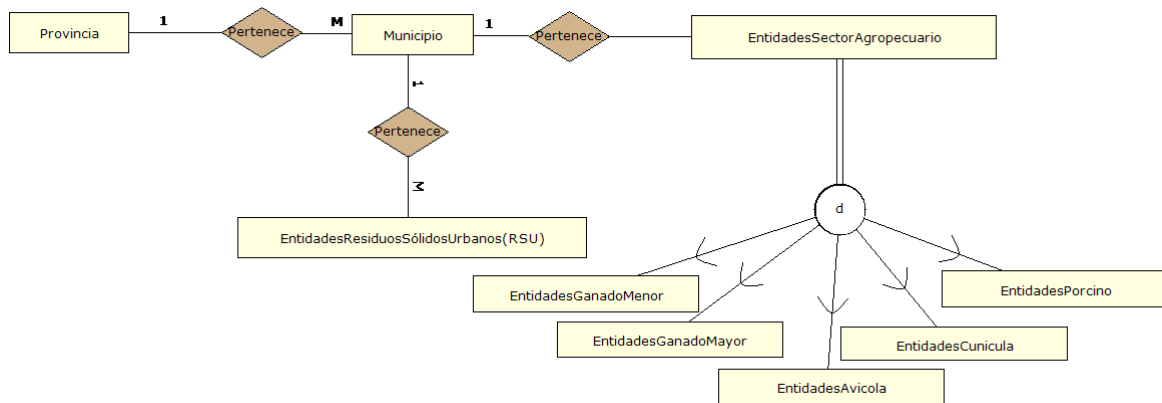
En la figura 2.3 se muestra el diagrama entidad relación para el sistema que gestiona las enti-

2.4 Fases del ciclo de migración para el diseño del sistema

Tabla 2.5: Actividades de la fase dos de MAADISH

Fase	Actividades
Migración total	<p>La migración total contempla el cambio total del sistema. Se migran todos los servicios y se diseñan herramientas: Se debe hacer en fases por sistema, donde se determina:</p> <ol style="list-style-type: none"> 1. Interfaces para gestionar entidades de residuos 2. Interfaz del Reporte General 3. Interfaces del potencial disponible e implementable 4. Interfaz de Geolocalización

Figura 2.3: Diagrama entidad relación del diseño de la base de datos del sistema uno: Agropecuario y RSU (residuos sólidos urbanos)



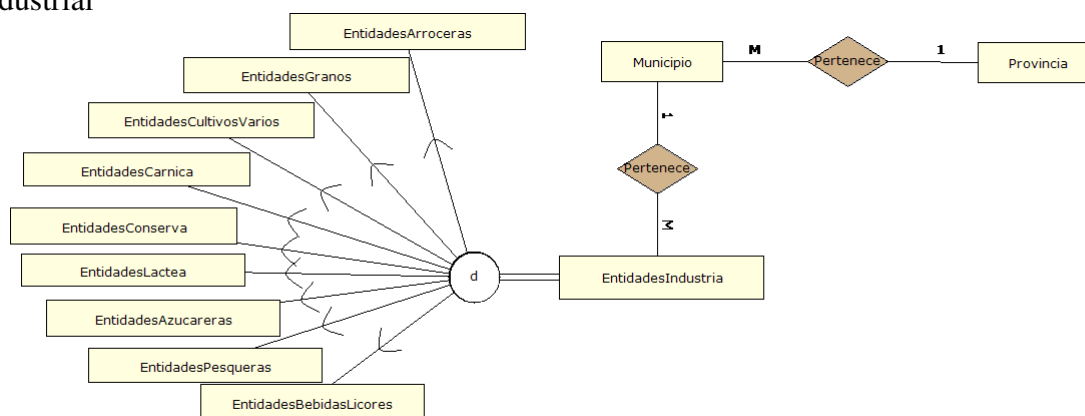
dades que generan residuos agropecuario y residuos sólidos urbanos.

En la figura 2.4 se muestra el diagrama entidad relación para el sistema que gestiona las entidades que generan residuos agroindustriales

A partir de estos diagramas entidad relación del diseño de las bases de datos de los sistemas anteriores, se crea, para el nuevo sistema migrado, una nueva base de datos donde se fundieron las bases de datos anteriores, manteniendo las funcionalidades y entidades en común, como son la provincia y el municipio. Pero para no repetir funcionalidades, ya que existen atributos en común entre entidades industria, entidades de residuos sólidos urbanos y entidades agropecuarias, se hace una nueva entidad que generalice los atributos de estas tres entidades, evitando la repetición de información que se hace innecesaria. Además se incorpora una nueva

2.4 Fases del ciclo de migración para el diseño del sistema

Figura 2.4: Diagrama entidad relación del diseño de la base de datos del sistema dos: Agroindustrial



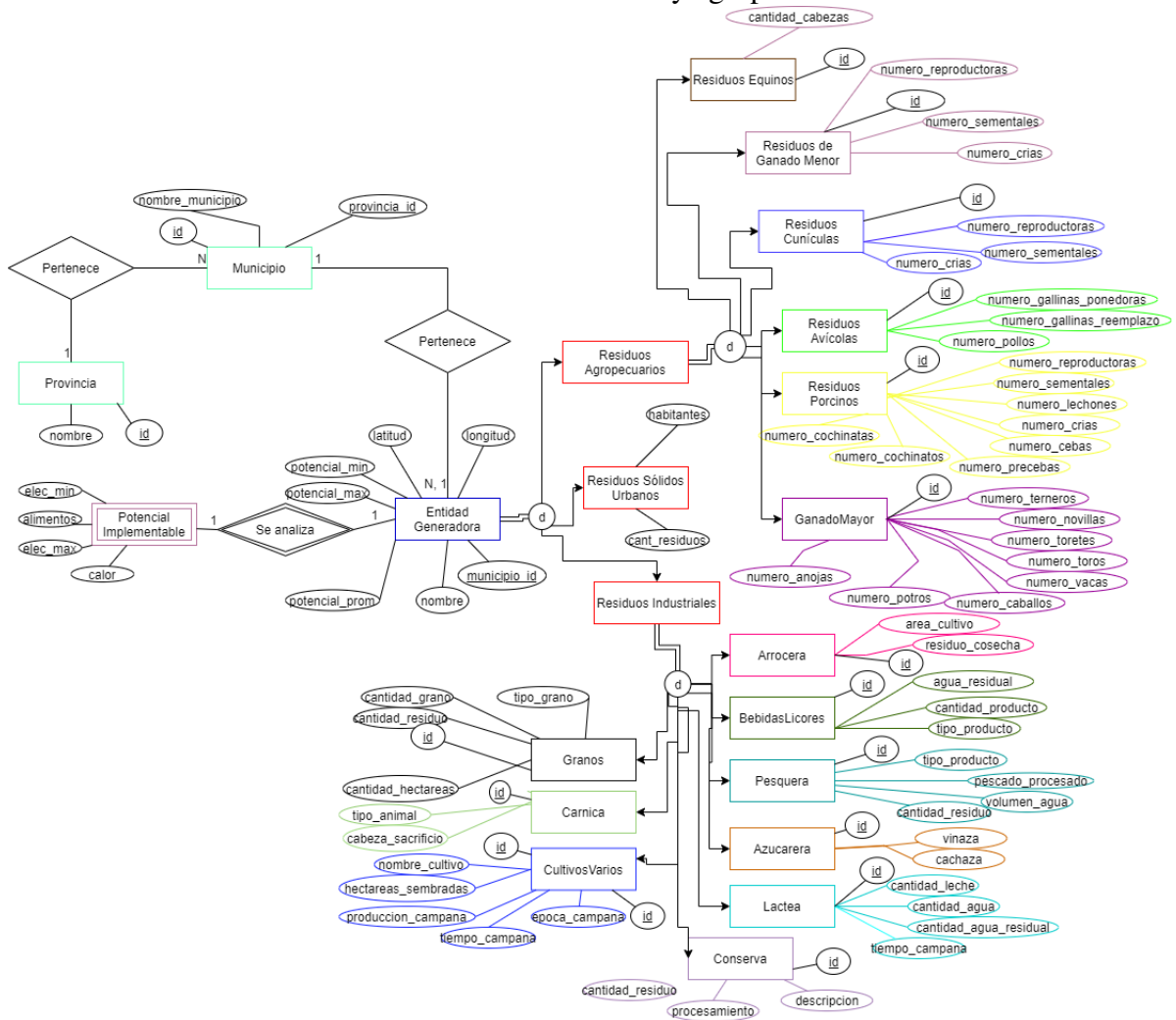
entidad agropecuaria a la base de datos: los residuos equinos, que tiene un cálculo diferente del potencial de biogás. Como se muestra a continuación todas las entidades tienen en común un nombre, una latitud, una longitud, un potencial mínimo, máximo y promedio, y un potencial implementable o también llamado uso final del biogás donde se calcula el potencial para la cocción de alimentos, los equipos de calor, la electricidad mínima y la electricidad máxima. Se creó una entidad generadora que incluye todos estos atributos y que pueden ser utilizados en las demás entidades haciendo más sencilla la migración de las bases de datos a una sola. Con el uso de la API REST fue necesario crear esta información nuevamente pero ya de forma más optimizada para su uso en el nuevo sistema.

Diseño de la Interfaz

Interfaces de Usuario Estos cálculos de los potenciales del biogás utilizados en los sistemas anteriores no eran exactamente los que se utilizan en este momento en el centro de estudios de la energía y los procesos industriales, sino que se hizo una nueva investigación sobre las fórmulas reales utilizadas, así como las coordenadas exactas en el mapa de las entidades que generan los residuos, y ahora, además de conocer con precisión el potencial disponible e implementable, se conoce además exactamente el lugar de la entidad generadora y de los nuevos tipos de residuos que se calculan de manera diferente.

2.4 Fases del ciclo de migración para el diseño del sistema

Tabla 2.6: Modelo entidad relación de la nueva base de datos a ser utilizada, integrando los modelos de bases de datos de los sistemas anteriores y agrupando entidades



2.4 Fases del ciclo de migración para el diseño del sistema

Tabla 2.7: Diseño de la interfaz



Tabla 2.8: Panel Antiguo de las aplicaciones



2.4 Fases del ciclo de migración para el diseño del sistema

Tabla 2.9: Panel de la aplicación actual

Especialista

Inicio

- Residuos que generan biogás
 - Residuos Porcinos
 - Residuos Cuniculas
 - Residuos Equinos
 - Residuos Avícolas
 - Residuos de Ganado Menor
 - Residuos Arroz
 - Residuos de Ganado Mayor
 - Residuos Azúcar
 - Residuos Bebidas y Licores
 - Residuos Pesca
 - Residuos Lácteos

Biogás

El biogás es un gas renovable compuesto principalmente por metano y dióxido de carbono obtenido a partir de la degradación anaerobia sin oxígeno de residuos orgánicos. Es, según el Instituto para la Diversificación y Ahorro de la Energía de España, "la única energía renovable que puede usarse para cualquiera de las grandes aplicaciones energéticas: eléctrica, térmica o como carburante". Se trata por tanto de transformar residuos ganaderos, agroindustriales y lodos de depuradoras de agua, pero también parte de los residuos domésticos. La basura se convierte así en la materia prima de una fuente de energía. .

El biogás es un gas renovable compuesto principalmente por metano y dióxido de carbono obtenido a partir de la degradación anaerobia sin oxígeno de residuos orgánicos. Es, según el Instituto para la Diversificación y Ahorro de la Energía de España, "la única energía renovable que puede usarse para cualquiera de las grandes aplicaciones energéticas: eléctrica, térmica o como carburante". Se trata por tanto de transformar residuos ganaderos, agroindustriales y lodos de depuradoras de agua, pero también parte de los residuos domésticos. La basura se convierte así en la materia prima de una fuente de energía. .

CEEPI

El Centro de Estudios de Energía y Procesos Industriales (CEEPI) de la Universidad de Sancti Spiritus, es un centro de estudios dedicado a realizar investigaciones, servicios científico técnicos y actividades de formación encaminadas a potenciar el desarrollo energético y agro-industrial sostenible en el territorio. Este centro lleva a cabo el programa general para el desarrollo local en el uso del biogás como fuente energética y de disminución de la contaminación ambiental, encaminado a desarrollar el proceso de producción de biogás en el territorio y su utilización eficiente con énfasis en la cocción de alimentos y la sustitución de combustibles fósiles en el sector agroindustrial, teniendo en cuenta los potenciales de residuos existentes, en aras de cerrar ciclo y contribuir al desarrollo socio económico del país.

El Centro de Estudios de Energía y Procesos Industriales (CEEPI) de la Universidad de Sancti Spiritus, es un centro de estudios dedicado a realizar investigaciones, servicios científico técnicos y actividades de formación encaminadas a potenciar el desarrollo energético y agro-industrial sostenible en el territorio. Este centro lleva a cabo el programa general para el desarrollo local en el uso del biogás como fuente energética y de disminución de la contaminación ambiental, encaminado a desarrollar el proceso de

Tabla 2.10: Login antiguo

[Inicio](#) [Iniciar sesión](#)

Iniciar sesión

Usuario

Contraseña







ACEPTAR

2.4 Fases del ciclo de migración para el diseño del sistema

Tabla 2.11: Login de la aplicación actual

2022 – Universidad de Sancti Spiritus José Martí

Tabla 2.12: Gestión de un residuo en la aplicación actual

Nombre Entidad	Numero Reproductoras	Numero Sementales	Numero Crias	Provincia	Municipio	Tipo Residuo	Latitud	Longitud	Acciones_del_usuario
EES ESASS UEB Agroindustrial Yaguajay	879	97687	6768	Sancti-Spiritus	Yaguajay	Industrial	22.321125041499158	-79.23188428382494	 
UEB OBE Yaguajay	768	98869	76	Sancti-Spiritus	Yaguajay	Agropecuario	22.321195234604406	-79.24440426456904	 
UEB Dos Rios Taguasco	8	8798	897	Sancti-Spiritus	Taguasco	Industrial	21.92945191102153	-79.2991932894523	 

Rows per page: 10 1-3 of 3

2022 – Universidad de Sancti Spiritus José Martí

Tabla 2.13: Gestión de un residuo en los sistemas anteriores

Entidades RSU

Provincia	Municipio	Nombre	Latitud	Longitud	
Sancti Spiritus	Trinidad	Campamento: Empresa MICONS	21.966997	-79.978608	POTENCIAL EDITAR ELIMINAR

2.5. Conclusiones parciales

Con la conclusión de este capítulo queda finalizada la parte del diseño y parte del desarrollo del nuevo sistema, teniendo ya el modelo de la nueva base de datos y de las interfaces del nuevo sistema, en comparación con los sistemas anteriores. Se definieron todos los requerimientos del nuevo sistema, tanto los que se mantienen como que se añaden más. Se trataron los factores a tener en cuenta para el desarrollo de la migración así como las fases del ciclo de vida de migración para el diseño y la implementación.

2.5 Conclusiones parciales

Tabla 2.14: Agregar un residuo en los sistemas anteriores/ Agregar un residuo en la aplicación actual

The image shows two overlapping screenshots of a web application. The top screenshot is a user registration form titled "Insertar Usuario". It includes a navigation bar with links: Inicio, Gestionar Usuario, Gestionar Provincia, Gestionar Municipio, Ayuda, and Cerrar Sesión. The form fields are: Usuario (empty), Roles (dropdown menu with "Admin" and "Especialista" options), Contraseña (password field with masked characters), Confirmar Contraseña (confirmation field), Nombre (empty), Apellidos (empty), and Correo (filled with "especialista"). A "REGRESAR AL LISTADO" button is located at the top left of the form area.

The bottom screenshot shows a mobile application interface. On the left is a sidebar menu with a user profile icon and the name "Especialista". The menu items are: Inicio, Residuos que generan biogás, Residuos Porcinos, Residuos Cuniculas, Residuos Equinos, Residuos Avícolas, Residuos de Ganado Menor, Residuos Arroz, Residuos de Ganado Mayor, Residuos Azúcar, Residuos Bebidas y Licores, Residuos Pesca, and Residuos Lácteos. The main screen displays a "Nueva Entidad" modal form. The form fields include: Nombre Entidad (empty), Provincia (dropdown menu with "Sancti-Spiritus" selected), Municipio (dropdown menu with options: Taguasco, Fomento, Cabaiguán, La Sierpe, Trinidad, Yaguajay), Numero Reproductoras (0), Numero de Ser (0), and latitud (0). Below the form, there is a "Calculando potencial..." section with values: Potencial min:0, Potencial max:0, Potencial promedio:0, Alimentos:0, Calor:0, Elec Min:0, and Elec Max:0. At the bottom of the modal are "CANCEL" and "SAVE" buttons. The background shows a table with columns for "Latitud" and "Longitud".

2.5 Conclusiones parciales

Tabla 2.15: Potenciales en las aplicaciones anteriores
Potencial Disponible de Biogás de Entidades Porcinas

REGRESAR AL LISTADO	
Provincia	Sancti Spiritus
Municipio	Trinidad
Nombre de la entidad	Ccahual
Latitud	21.854
Longitud	-79.547
Número de reproductoras	123
Número de sementales	3
Número de cochinos	3
Número de cochinas	3
Número de cebas	123
Número de lechones	3
Número de precebas	123
Número de crías	3
Potencial Disponible Mínimo	49.46238
Potencial Disponible Máximo	59.9544

Tabla 2.16: Vista de los potenciales en la aplicación actual

Entity Name	Province	Municipality	Number of Reproductors	Number of Sementales	Number of Pigs	PD Min	PD Max	PD Prom	Entity Type
EES ESASS UEB Agroindustrial Yaguajay	Sancti-Spiritus	Yaguajay	879	97687	6768	36451.81	44184.01	40317.91	Agropecuaria
UEB Dos Rios Taguasco	Sancti-Spiritus	Taguasco	7	8798	897	3272.86	3967.1	3619.98	Agropecuaria
UEB OBE Yaguajay	Sancti-Spiritus	Yaguajay							

Tabla 2.18: Perfil *backend* y *frontend* para la vista del mapa

```
    async getLocations() {
      var locations = [];
      await axios.all([
        await axios.get('http://localhost:1337/api/arroceras').then((response) => {
          const arroceras = response.data.data
          arroceras.forEach(item => {
            locations.push({
              id: item.id,
              coord: [item.attributes.latitud, item.attributes.longitud],
              info: `
                <p>
                  Entidades Arroceras
                  <br/>
                  Nombre: ${item.attributes.nombre_entidad}
                  <br/>
                  Potencial Biogás Promedio: ${item.attributes.pdprom}
                  <br/>
                  Potencial Biogás Cocción de Alimentos: ${item.attributes.alimentos}
                  <br/>
                  Potencial Biogás Equipos de Calor: ${item.attributes.calor}
                  <br/>
                  Potencial Biogás Electricidad Mínima: ${item.attributes.elec_min}
                  <br/>
                  Potencial Biogás Electricidad Mínima: ${item.attributes.elec_max}
                </p>`
            })
          })
        })
      ])
    }
  }
}

<template>
  <v-container class="cont" elevation="18">
    <l-map elevation="18" :center="center" :zoom="zoom" class="map" ref="map" @update:center="centerUpdated"
      @update:zoom="zoomUpdated">
      <l-tile-layer :url="url">
      </l-tile-layer>
      <l-marker v-for="marker in markers" :key="marker.id" :lat-lng="marker.coord" :icon="icon">
        <l-icon :icon-size="dynamicSize" :icon-anchor="dynamicAnchor" icon-url="marker.png">
        </l-icon>
        <l-tooltip :content="marker.info">
        </l-tooltip>
      </l-marker>
    </l-map>
  </v-container>
</template>
<script>
import L from 'leaflet';
import { LMap, LTileLayer, LMarker, LIcon, LTooltip } from 'vue2-leaflet';
import axios from 'axios';
```

Tabla 2.19: Perfil *backend* y *frontend* para el inicio de sesión

```
async PostUser() {
  try {
    const {data}=await axios.post('http://localhost:1337/api/auth/local', {
      identifier: this.identifier,
      password: this.password,
    })

    console.log(data);
    sessionStorage.setItem('jwt', data.jwt)
    sessionStorage.setItem('username', data.user.username)
    sessionStorage.setItem('email', data.user.email)
    sessionStorage.setItem('rol', data.user.rol)

    location.reload()

  } catch(error) {
    this.error = true
  }
}

</v-alert>
<v-text-field
  :rules="[rules.required, rules.length(1)]"
  v-model="identifier"
  filled
  color="primary"
  label="nombre usuario"

  placeholder="nombre de usuario"
  prepend-inner-icon="mdi-account"
/>
<v-text-field
  v-model="password"
  :rules="[ rules.required]"
  filled
  color="primary"
  counter="9"
  placeholder="Contraseña"
  prepend-inner-icon="mdi-key"
  label="Contraseña"
  style="min-height: 96px"
  :type="passwordShow ? 'text': 'password'"
  :append-icon="passwordShow ? 'mdi-eye': 'mdi-eye-off'"
  @click:append="passwordShow = !passwordShow"
/>
```

Tabla 2.20: Menú lateral del *frontend*

```
<template >
  <v-app id="main" >

    <v-navigation-drawer
      v-model="drawer"
      v-if="nombre_usuario"
      :clipped="clipped"
      fixed
      app
      width="370"
      color="blue lighten-1">

      <div class="text-center" >
        <v-avatar size="80" color="primary" size="60" class="mt-3" v-if="nombre_usuario">
          <v-icon size="70" color="white">mdi-account</v-icon>
        </v-avatar>

        <h1 class="text-h6 ml-5" mt="8" >{{nombre_usuario}}</h1>
      </div>

      <v-list>
        <v-list-item
          to="/" >
          <v-list-item-icon>
            <v-icon color="primary">mdi-view-dashboard</v-icon>
          </v-list-item-icon>

          <v-list-item-title>Inicio</v-list-item-title>
        </v-list-item>
      </v-list>
    </v-navigation-drawer>
  </v-app>
</template>
```

entidades:

```
{ title: 'Residuos Porcinos', icon: 'mdi-pig', to: '/agropecuario/porcino' },
{ title: 'Residuos Cuniculas', icon: 'mdi-rabbit', to: '/agropecuario/cuniculas' },
{ title: 'Residuos Equinos', icon: 'mdi-horse', to: '/agropecuario/equino' },
{ title: 'Residuos Avícolas', icon: 'mdi-bird', to: '/agropecuario/avicolas' },
{ title: 'Residuos de Ganado Menor', icon: 'mdi-sheep', to: '/agropecuario/ganado-menor' },
{ title: 'Residuos Arroz', icon: 'mdi-rice', to: '/industrial/arroceras' },
{ title: 'Residuos de Ganado Mayor', icon: 'mdi-cow', to: '/agropecuario/ganado-mayor' },
{ title: 'Residuos Azúcar', icon: 'mdi-spoon-sugar', to: '/industrial/azucarera' },
{ title: 'Residuos Bebidas y Licores', icon: 'mdi-glass-wine', to: '/industrial/bebidas-licores' },
{ title: 'Residuos Pesca', icon: 'mdi-fish', to: '/industrial/pesquera' },
{ title: 'Residuos Lácteos', icon: 'mdi-cup', to: '/industrial/lactea' },
{ title: 'Residuos de Conserva', icon: 'mdi-fridge', to: '/industrial/conserva' },
```

CAPÍTULO 3

Implementación y pruebas

Introducción

En el presente capítulo se mostrarán las fases que llevan al desarrollo de la aplicación que migra los sistemas, haciendo una continuación de las fases comenzadas en el capítulo anterior. Se hace una descripción de los datos generales del proyecto y del software a migrar.

3.1. Datos generales del proyecto

Nombre de la Institución: Centro de estudios de la energía y los procesos industriales (CEEPI) En colaboración con la Universidad de Sancti-Spíritus

Nombre del Proyecto: Migración de sistemas relacionado al cálculo del potencial del biogás en entidades que generan residuos.

DATOS DEL SISTEMA /SOFTWARE A MIGRAR **Nombre versión del sistema /software a migrar:** Sistemas heredados para el cálculo del potencial de producción del biogás.

3.2 Fases del ciclo de migración para el desarrollo y la implementación del sistema

Alternativa de software de código abierto a implementar: IDE Visual Studio Code: No-dejs

De metodologías mencionadas anteriormente se describe con la seleccionada para el desarrollo durante todo el proceso de migración del Sistema para el cálculo del potencial del Biogás.

3.2. Fases del ciclo de migración para el desarrollo y la implementación del sistema

3.2.1. Fase 3: Pruebas y resultados

Se implementa un ambiente de pruebas lo más semejante al ambiente de producción en donde se pueda ejecutar un prototipo con información actualizada; si son los resultados esperados se podrá continuar con la siguiente fase, de no serlo se deberá regresar al análisis y esquematización de entradas y salidas de datos. Es aquí otra de las características de MADIISH, permite regresar dentro del ciclo de migración a un punto en el que se pueda determinar las fallas presentadas según el avance obtenido (Power Data, 2018)

La tabla 3.1 muestra las actividades que se realizan en esta fase

Propuestas de Pruebas

1. Pruebas propuestas para el *login*

Validaciones:

1. El campo usuario debe estar registrado previamente en la base de datos
2. La contraseña debe estar registrada asociada a un usuario en la base de datos

En la tabla 3.2 se muestra el formato de planeamiento de pruebas para el *login* con el uso de MAADISH

3.2 Fases del ciclo de migración para el desarrollo y la implementación del sistema

Tabla 3.1: Actividades de la fase 3 de MAADISH

Fase	Actividades
Documentación de la migración	<ol style="list-style-type: none">1. Se deberá documentar todo el proceso paso a paso resaltando las experiencias que se consideren relevantes y que puedan ser de utilidad durante las migraciones.2. Se deberá documentar todas las pruebas realizadas en el laboratorio (pruebas de hardware y pruebas de software), de manera que pueda ser utilizada como material de apoyo y/o referencia.3. La documentación de estas pruebas permitirá elaborar un manual de procedimientos y/o protocolos de pruebas, para usar en el laboratorio.4. La documentación deberá realizarse en un formato estándar.
Capacitación de usuarios	Capacitación del usuario final

2. Pruebas propuestas para registrar un nuevo usuario

Validaciones:

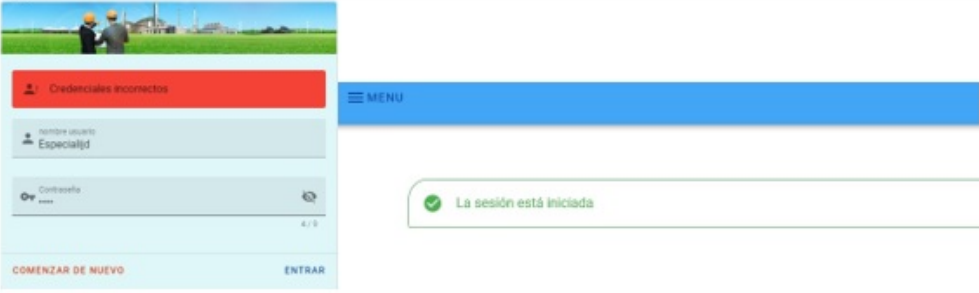
1. El campo usuario debe ser alfabético principalmente, pero debe aceptar números enteros
2. El campo email debe contener letras minúsculas, sin caracteres especiales y sin espacios entre las letras, conteniendo el carácter @ y el nombre del servidor o el dominio si contiene
3. El campo contraseña se le va a permitir no más de 9 caracteres, exigiendo en ellos letras mayúsculas y minúsculas, caracteres especiales y números
4. Los campos email y username deben ser únicos

En la tabla 3.3 se muestra el plan de prueba para el registro de un nuevo usuario al sistema

3. Pruebas propuestas para agregar una nueva entidad

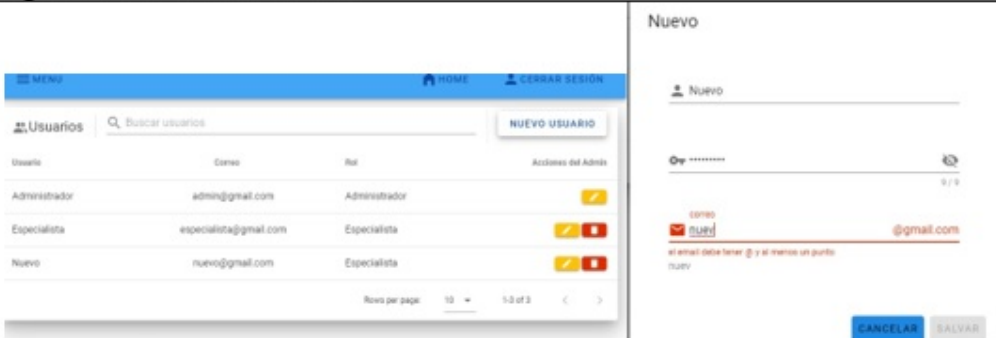
3.2 Fases del ciclo de migración para el desarrollo y la implementación del sistema

Tabla 3.2: Formato de pruebas para el *login*

Formato Plan de Pruebas de Usuario			
Identificador	1	Versión	1
Responsable	Angelica Rivero		
Nombre del Caso de Prueba	Login		
Módulo	Submódulo		
Ingreso del usuario al sistema			
Formulario			
Pages/login.vue			
Descripción de la prueba : El usuario se autentica en el sistema			
Datos del user:			
Username: Especialista			
Contraseña: Especialista*123			
Resultados esperados: Sesión iniciada			
Resultados reales: Al entrar los datos erróneos se mostró una alerta de “credenciales incorrectos”, al entrar los datos correctos se mostró una alerta de sesión iniciada, en caso de poner cualquier valor en la contraseña no se mostró el botón de entrar al sistema, mostrándose un info con los caracteres que debe llevar en ese campo			
Usuario logueado correctamente			
<p>Imagen:</p>  <p>The image shows a web application interface for user login. On the left, there is a login form with a red error message at the top that reads 'Credenciales incorrectas'. Below the error message, the form shows the username 'Especialistd' and a password field with masked characters. At the bottom of the form, there are two buttons: 'COMENZAR DE NUEVO' and 'ENTRAR'. On the right side of the screen, there is a blue navigation bar with a 'MENU' icon. Below the navigation bar, there is a green success message that reads 'La sesión está iniciada'.</p>			

3.2 Fases del ciclo de migración para el desarrollo y la implementación del sistema

Tabla 3.3: Formato plan de pruebas para el registro de un nuevo usuario

Formato Plan de Pruebas de Usuario			
Identificador	2	Versión	1
Responsable	Angelica Rivero		
Nombre del Caso de Prueba	Usuario		
Módulo	Submódulo		
Registro de un nuevo usuario en el sistema			
Formulario			
Pages/para_admin/usuarios.vue			
Descripción de la prueba : Se ingresa un nuevo usuario al sistema			
Datos del user:			
Nombre del usuario: administrador, especialista			
Email: admin@gmail.com , especialista@gmail.com			
Contraseña: Admin*123, Especialista*123			
Resultados esperados: Se registra el usuario en el sistema			
Resultados reales: Al entrar datos correctos al sistema, se registra el usuario y se muestra en la tabla de usuarios, de lo contrario el sistema no deja mostrar el botón de salvar los cambios			
Usuario registrado correctamente			
<p>Imagen:</p>  <p>The image shows two parts of a web application. On the left is a table titled 'Usuarios' with columns for 'Usuario', 'Correo', 'Rol', and 'Acciones del Admin'. It lists three users: 'Administrador' (admin@gmail.com), 'Especialista' (especialista@gmail.com), and 'Nuevo' (nuevo@gmail.com). On the right is a 'Nuevo' form with fields for 'Nombre' (containing 'Nuevo'), 'Correo' (containing 'nuevo@gmail.com'), and 'Contraseña' (containing 'nuevo'). There are 'CANCELAR' and 'SALVAR' buttons at the bottom of the form.</p>			

3.2 Fases del ciclo de migración para el desarrollo y la implementación del sistema

Validaciones:

1. Las cantidades de residuos no deben ser negativas, y sólo deben ser numéricas, sin caracteres especiales
2. El nombre de la entidad debe ser alfabético pero puede incluir números y caracteres especiales
3. La latitud no debe exceder los 90 grados
4. La longitud no debe exceder los 180 grados

4. Pruebas a la geolocalización (precisión del mapa): Se prueba la localización de una entidad y la exactitud de la misma en el mapa

3.2.2. Fase 4: Implementación

Una vez superadas las fases anteriores es necesario implementar el nuevo sistema en el ambiente de producción por lo que se debe anular el funcionamiento del subsistema heredado migrado, actualizar la base de datos del nuevo sistema e implementar finalmente el nuevo sistema.


Implementación de las pruebas planeado en la fase anterior al nuevo sistema

Pruebas automatizadas en respuesta al planeamiento Para implementar las pruebas planificadas al nuevo sistema se utiliza la herramienta integrada del Chrome Selenium, que es una extensión que guarda las acciones del usuario como una secuencia de pasos y luego los repite automáticamente, y , si se desea cambiar datos se pueden agregar en la secuencia de pasos guardada. Este proceso fue aplicado en los módulos de inicio de sesión, luego de iniciar la sesión se realizaron las pruebas al añadir un nuevo usuario al sistema, y luego para el proceso de insertar, modificar, eliminar o buscar una entidad en el sistema.

En la figura 3.1 se muestran los resultados chequeados de manera satisfactoria en cada paso de la secuencia guardada

3.2 Fases del ciclo de migración para el desarrollo y la implementación del sistema

Tabla 3.4: Formato plan de pruebas para agregar una nueva entidad al sistema

Formato Plan de Pruebas de Entidad (Ganado Menor)			
Identificador	3	Versión	1
Responsable	Angelica Rivero		
Nombre del Caso de Prueba	Usuario		
Módulo	Submódulo		
Registro de un nuevo usuario en el sistema			
Formulario			
Pages/agropecuario/ganadomenor.vue			
Descripción de la prueba : Se ingresa un nuevo usuario al sistema			
Datos de la entidad: (datos correctos e incorrectos con fin de probar)			
Nombre Entidad: La nueva			
Número Reproductoras: 67.89			
Número Sementales: 839.98372			
Número Crías: 98309.873			
Latitud: 432			
Longitud: 432524			
Resultados esperados: Se registra la entidad en el sistema			
Resultados reales: Al entrar datos correctos al sistema, se registra la entidad, de lo contrario el sistema no deja mostrar el botón de salvar los cambios			
Entidad registrada correctamente			
<p>Imagen:</p> 			

3.2 Fases del ciclo de migración para el desarrollo y la implementación del sistema

Tabla 3.5: Organización del nuevo sistema en cuanto a módulos y servidores

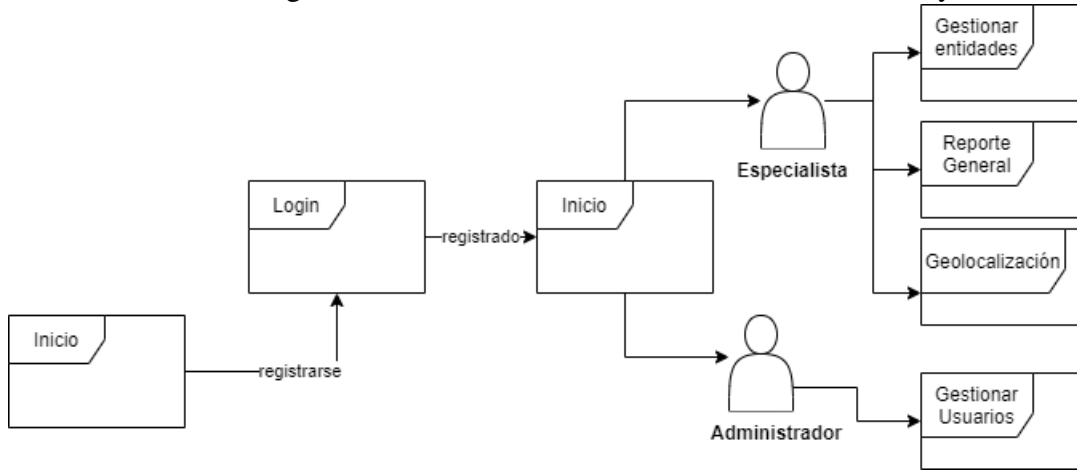
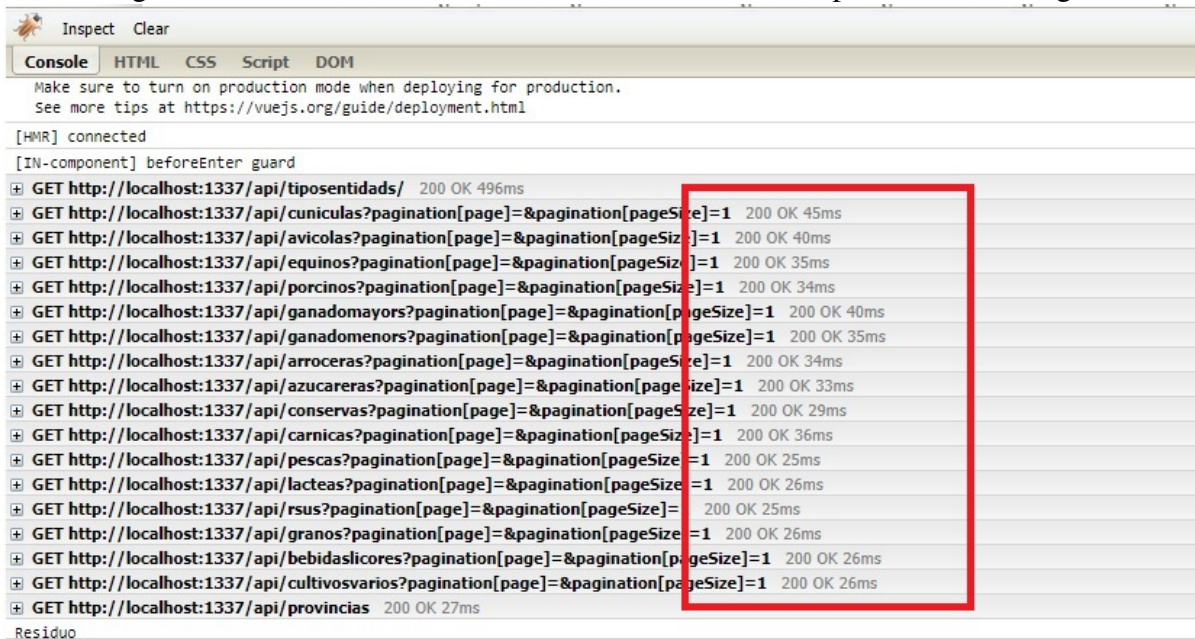


Figura 3.1: Secuencia de pasos realizados en Selenium de manera exitosa para añadir una nueva entidad en el sistema, pasando primero por el inicio de sesión

Command	Target	Value
3 ✓ click	id=input-31	
4 ✓ type	id=input-31	Especialista
5 ✓ click	css=v-icon-link	
6 ✓ type	id=input-35	Especialista123*
7 ✓ click	css=white-text > .v-btn__content	
8 ✓ click	css=v-toolbar__content > .v-btn:nth-child(1) > .v-btn__content	
9 ✓ click	css=v-list-item:nth-child(1) > .v-list-item__content	
10 ✓ click	css=mb-2:nth-child(8)	
11 ✓ click	id=input-169	
12 type	id=input-169	Nuevo

3.2 Fases del ciclo de migración para el desarrollo y la implementación del sistema

Figura 3.2: Pruebas de rendimiento realizadas al sistema para los métodos get



Pruebas de rendimiento Para realizar las pruebas de rendimiento se incluye el uso de la extensión FireBug, instalada en el navegador chrome, que funciona como el inspector, mostrando el tiempo de respuesta de cada petición.

En la figura 3.2 se muestran los tiempos de respuesta de cada petición hecha al sistema.

En la figura 3.3 se muestran las pruebas de rendimiento para el método delete

Pruebas de funcionalidad Para realizar las pruebas de funcionalidad, primeramente se visualiza cada interfaz del sistema, verificando si se ve bien, si se pueden acceder bien a cada módulo, se verifica si cumple con las expectativas del cliente de cómo debe ser cada interfaz, y si cumple con las funcionalidades requeridas.

Primeramente se muestra la interfaz de Agregar una entidad al sistema.

Se muestra que la información dentro del cuadro puede estar más centrada, más organizada, y los botones pudieran tener una mejor visualización, la interfaz cumple con las expectativas de funcionalidad del cliente, quien espera que al no poner los datos correctos en los respectivos campos, se muestre información en rojo diciendo lo que se necesita en ese cuadro y lo que

3.2 Fases del ciclo de migración para el desarrollo y la implementación del sistema

Figura 3.3: Pruebas de rendimiento para el método delete

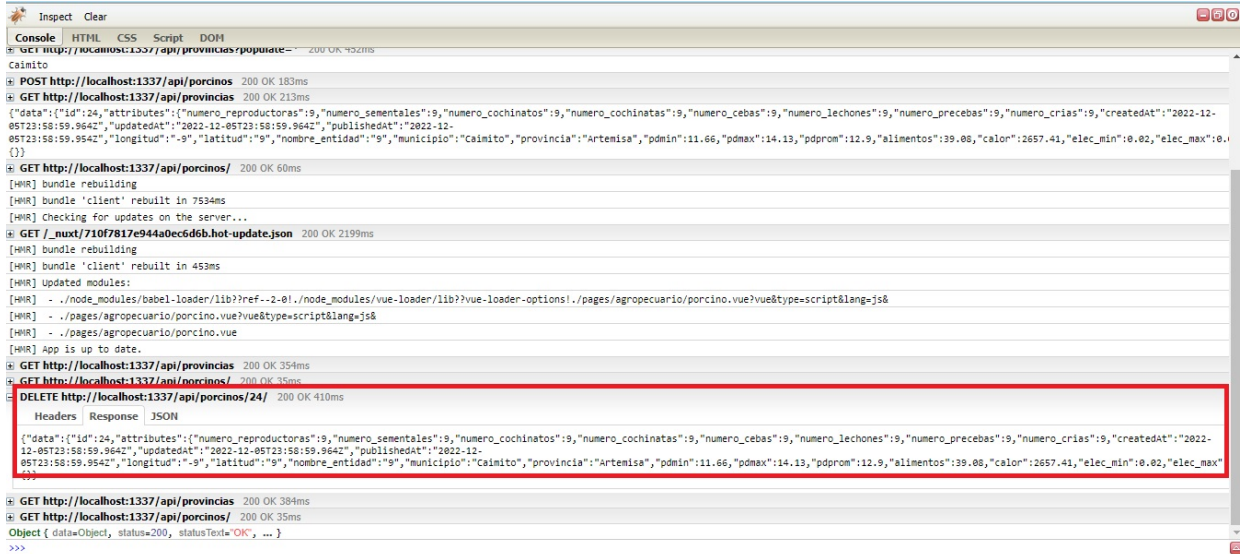
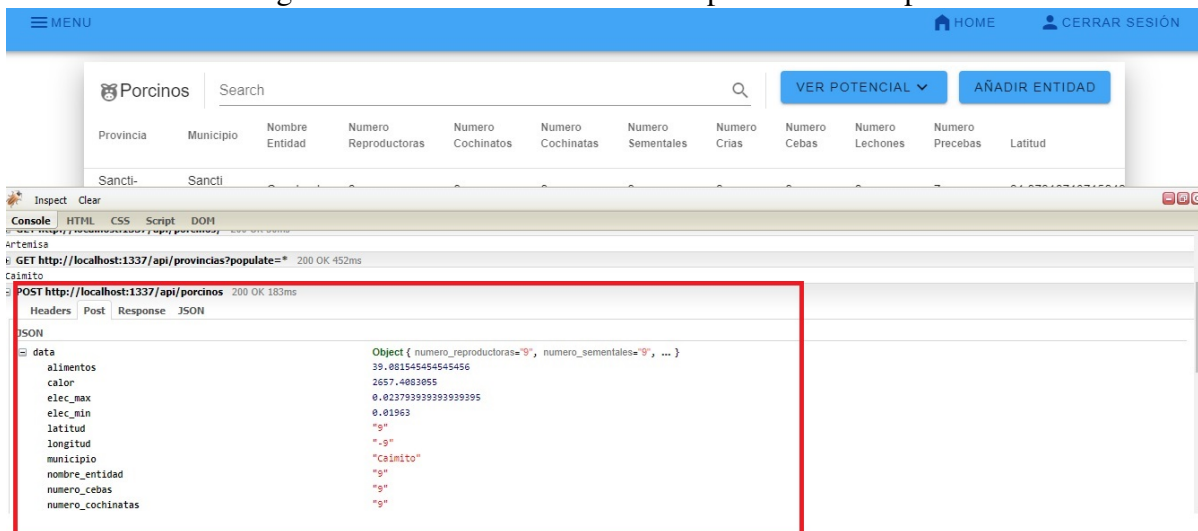


Figura 3.4: Pruebas de rendimiento para el método post



3.2 Fases del ciclo de migración para el desarrollo y la implementación del sistema

Figura 3.5: Prueba de rendimiento y funcionalidad realizada a los filtros del reporte

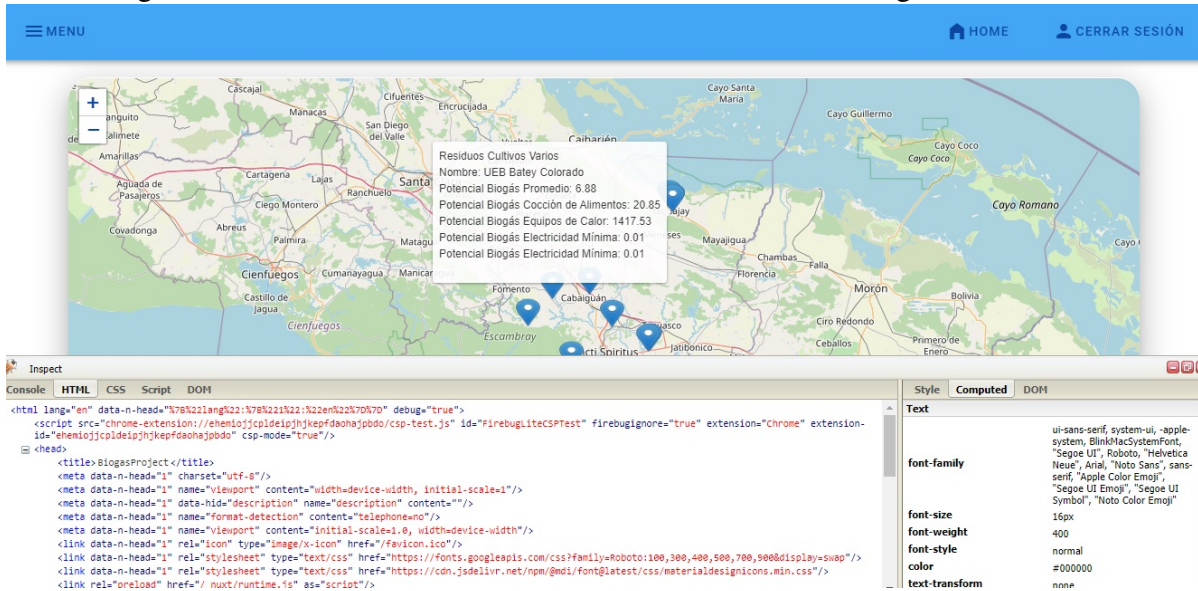
Tipo Residuo	Tipo Entidad	Provincia	Municipio	Nombre entidad	Potencial Promedio	Cocción Alimentos	Equipos de Calor	Electricidad Mínima	Electricidad Máxima
Cunicula	Agropecuaria	Sancti-Spiritus	Taguasco	UEB Dos Rios Taguasco	3619.98	10969.63	745896.62	5.51	6.68

Figura 3.6: Prueba de rendimiento realizada a los filtros del reporte por tipo de entidad

Tipo Residuo	Tipo Entidad	Provincia	Municipio	Nombre entidad	Potencial Promedio	Cocción Alimentos	Equipos de Calor	Electricidad Mínima	Electricidad Máxima
Agropecuaria	Agropecuaria	Sancti-Spiritus	Taguasco	UEB Dos Rios Taguasco	3619.98	10969.63	745896.62	5.51	6.68

3.2 Fases del ciclo de migración para el desarrollo y la implementación del sistema

Figura 3.7: Prueba de funcionalidad realizada a la interfaz de geolocalización



puso incorrectamente el usuario. Además se espera que, si se pone información incorrecta, el sistema no deje insertar ninguna entidad, por lo que en respuesta, no se mostró el botón salvar, o sea no se habilitó hasta que el usuario no escribió bien todos los datos.

Continuando con estas pruebas, se muestra la interfaz del reporte general, que anteriormente tenía una paginación del lado del cliente, por lo que se cambió del lado del servidor, y además tiene un botón de exportar a excel que tiene la funcionalidad de tomar los datos que se filtren desde la interfaz a la base de datos para luego ser guardados en un arreglo y exportados a un excel. Además, el cliente sugirió que fueran hechos filtrados de tipo de residuo, tipo de entidad, provincia y municipio. A continuación se muestra la interfaz de los reportes con el filtrado realizado y luego ese mismo filtro exportado a excel:

Como se puede observar, aunque el cliente no vea todos los elementos filtrados porque está en la primera página que viene del lado del servidor, al exportar los filtros en el excel sí se muestra toda la información que viene de la base de datos.

3.2 Fases del ciclo de migración para el desarrollo y la implementación del sistema

Figura 3.8: Agregando una entidad al nuevo sistema

La Nueva

seleccione una provincia
Artemisa X ▾

seleccione un municipio
Caimito X ▾

	Numero Sementales	Numero Crias
Numero Repro...	839.98372	98309.873

El campo debe existir y ser no negativo

latitud	longitud
432	432524

La latitud debe existir, ser menor que 90 y no debe contener letras
Calculando potencial...
Potencial min:NaN
Potencial max:NaN
Potencial promedio:NaN
Alimentos:NaN
Calor:NaN
Elec Min:NaN
Elec Max:NaN

La latitud debe existir, ser menor que 180 y no debe contener letras

CANCEL SAVE

3.2 Fases del ciclo de migración para el desarrollo y la implementación del sistema

Figura 3.9: Filtrados en la interfaz con el paginador *server side* a elemento por tipo de residuo

The screenshot shows a web application interface with a blue header containing a menu icon and links for 'HOME' and 'CERRAR SESIÓN'. Below the header, there are two filter dropdowns: 'Agropecuaria' and 'Cunícula'. A third filter dropdown is set to 'Sancti-Spíritus' with a 'Filtrar por Municipio' dropdown next to it. Below the filters is a table with the following data:

Tipo Residuo	Tipo Entidad	Provincia	Municipio	Nombre entidad	Potencial Promedio	Cocción Alimentos	Equipos de Calor	Electricidad Mínima	Electricidad Máxima
Cunícula	Agropecuaria	Sancti-Spíritus	Taguasco	UEB Dos Ríos Taguasco	3619.98	10969.63	745896.62	5.51	6.68

Below the table, there are navigation arrows and the text 'Se han encontrado 3 elementos en total' and 'Page: 1'.

Figura 3.10: Documento excel con todos los elementos que encontró la base de datos a partir de esos filtros

The screenshot shows an Excel spreadsheet window titled 'EntidadesBiogas. (1).xls - Excel' overlaid on the web application interface. The spreadsheet contains the following data:

	A	B	C	D	E	F	G	H	I	J
1	provincia	municipio	tipo_entidad	pdprom	alimentos	calor	elec_min	elec_max	tipo_residuo	
2	Sancti-Spíritus	Taguasco	Agropecuaria	3619.98	10969.63	745896.62	5.51	6.68	Cunícula	
3	Sancti-Spíritus	Yaguajay	Agropecuaria	40317.91	122175.48	8307505.02	61.37	74.38	Cunícula	
4	Sancti-Spíritus	Yaguajay	Agropecuaria	8	76	768	78	87	Cunícula	
5										
6										
7										
8										
9										
10										
11										

The Excel window also shows a status bar at the bottom with 'Listo' and a zoom level of 126%.

3.2.3. Fase 5: Migración continua y documentación

Se realiza una documentación del sub sistema o sistema migrado según sea el caso en la cual se describen los procesos de negocios, la documentación del código fuente y la descripción gráfica del sistema (Pumasunta Chungandro, 2020).

3.2.4. Fase T: Elaboración de la documentación final

Elaboración de la documentación final, consiste en unir toda la documentación y retroalimentar los manuales o procesos en los que se encuentre ambiguo su contenido(Pumasunta Chungandro, 2020).

3.3. Conclusiones parciales

Con la conclusión del capítulo se desarrolló, implementó y se hicieron pruebas de aceptación del usuario o de funcionalidad resultando todas las pruebas exitosas . Además se realizó la documentación del sistema migrado con los procesos de negocio, la documentación del código fuente y la descripción gráfica del sistema. Se hizo un manual de usuario y se actualizó la ayuda del sistema.

CONCLUSIONES

1. En el desarrollo de esta investigación se presentó un marco teórico, el cual permitió conocer más acerca del significado de: migración, software heredado, migración de software heredado, y tipos de migraciones de software heredado, así como las razones por las cuales se utilizó la metodología MADIISH. Además, pudimos tener una mayor idea del significado, la producción y la utilización del biogás, así como la geolocalización necesaria para ver las entidades que generan los residuos en el mapa. También quedaron definidas las herramientas y tecnologías utilizadas, que son novedosas en el contexto actual y que facilitaron la migración de los sistemas, así como el mejoramiento de las interfaces y las funcionalidades. Se concluyó que realmente era necesaria la migración de los sistemas a uno nuevo.
2. En el segundo capítulo de esta investigación se definieron los factores a tener en cuenta para el desarrollo de la nueva aplicación como: las reglas del negocio, los requisitos funcionales y no funcionales, las buenas prácticas y por último las fases del ciclo de migración de esta nueva metodología, donde se vio el cambio de funcionalidades e interfaces de las aplicaciones anteriores a la aplicación actual.
3. En el desarrollo del último capítulo se vieron reflejadas las fases de implementación y pruebas realizadas al nuevo sistema, las cuales fueron exitosas, realizándose pruebas de funcionalidad, de rendimiento y de aceptación del usuario, y en respuesta a las pruebas que planeó el cliente, se realizaron pruebas automáticas al software resultando exitosas. Se obtuvo el nuevo sistema migrado con nuevas funcionalidades añadidas, nuevos residuos de biogás que se calculan de manera diferente, y una mayor rapidez en la in-

teracción con las interfaces de usuario, mostrándose la información necesaria de los potenciales de biogás en el mapa (Jiménes, 2021).

RECOMENDACIONES

Se recomienda

- Utilizar la metodología de esta investigación para futuras investigaciones cuando se necesite hacer una actualización o un cambio de software en una entidad y no se conozca el procedimiento a llevar para esta actualización.

REFERENCIAS

- Almeida, J. (2022). Node js vs. PHP: An In-depth Comparison Guide for Web Development.
- Beltrán López, G. (2016). *Geolocalización online: La importancia del dónde*.
- Castillo Oporto, R. y Figueredo Denis, L. J. (2020). *Estudio comparativo de metodologías de desarrollo de software con base a la migración de sistemas heredados*, PhD thesis.
- de Compostaje, R. E. (2017). *Ingeniería y Aspectos Técnicos de la Digestión Anaeróbica*, Vol. II. 4.
- García, J. (2021). Qué es la migración de Software?
- Github (2022). The top programming languages.
URL: *The top programming languages TheStateoftheOctoverse*
- Hurtado, E, M. (2013). *El biogás puede mejorar la sustentabilidad del agro en América Latina*.
- Jiménes, J. (2021). *METODOLOGÍA PARA EL DISEÑO DE PROGRAMAS PARA LA PRODUCCIÓN Y APROVECHAMIENTO DEL BIOGÁS A NIVEL LOCAL Y TERRITORIAL*.
- Llamas, L. (2019). VUETIFY, ESTÉTICA MATERIAL DESIGN PARA TUS APPS EN VUEJS.
- MD, E. (2015). Proyecto de biogas en PYMES del sector agropecuario.
- Menendez, R., B. A. (2017). Ingeniería del software: Metodologías de desarrollo, Informática Aplicada a la Gestión Pública.

- Ochoa, M. (2022). 12 lenguajes de programación con demanda en 2022 (índice TIOBE – febrero 2022).
URL: <https://www.itmastersmag.com/noticias-analisis/12-lenguajes-de-programacion-con-demanda-en-2022/>
- Pérez Porto, J. y Merino, M. (2022). Definición de geolocalización - Qué es, Significado y Concepto. Definicion.de.
- Power Data, E. e. G. d. d. (2018). *Introducción a la Migración de datos*.
- Pumasunta Chungandro, E. S. (2020). *MIGRACIÓN DE SISTEMA “CONTROL DE BIENES” DE ORACLE FORMS A UNA ARQUITECTURA (JSF) JAVA PARA LA EMPRESA MUNICIPAL DE MOVILIDAD Y OBRAS PÚBLICAS (EMMOP-Q)*, PhD thesis, UNIVERSIDAD TECNOLÓGICA ISRAEL, Quito, Ecuador.
- Red, H. (2020). ¿Qué es una API de REST?
URL: <https://www.redhat.com/es/topics/api/what-is-a-rest-api>
- Russi, Á. (2022). Desarrollo web, Programación.
- Sánchez (2016). *Biogás en el mundo*.
- Sánchez, A. (2021). Definición de geolocalización. Retrieved from Xinhua. (2013). Biogás gana espacio como fuente de energía en Cuba.
URL: www.americaeconomia.com Zinder, S. H.
- Semah, B. (2022). What Exactly is Node.js? Explained for Beginners.
URL: <https://www.freecodecamp.org/news/what-is-node-js/>
- Skaftei, F. y Angelidaki, I. (2022). *Biological Strategies to Enhance the Anaerobic Digestion Performance: Fundamentals and Process Development*.
- Spiegato (2022). Migración de sistemas y códigos heredados.
- Suárez-Hernández, J. (2020). *Evaluación del potencial de producción del biogás en Cuba*.
- Tarazona, A. Y. L. (2017). Pautas para una correcta Migración de Bases de Datos.
- Varnero Moreno, M. T. (2012). *Manual de Biogás*.
- Wagner, C. (2014). *Model-Driven Software Migration*.

ANEXO A

Estadísticas

En febrero de este año, los 12 lenguajes de programación más populares en el índice TIOBE fueron:

Python, con un rating de 15.33 %

C, con 14.08 % Java, con 12.13 %

C++, con 8.01 %

C#, con 5.37 %

Visual Basic, con 5.23 %

JavaScript, con 1.83 %

PHP, con 1.79 %

Assembly language, con 1.60 %

SQL, con 1.55 % Go, con 1.23 %

Swift, con 1.18 %

Figura A.1: Índice TIOBE

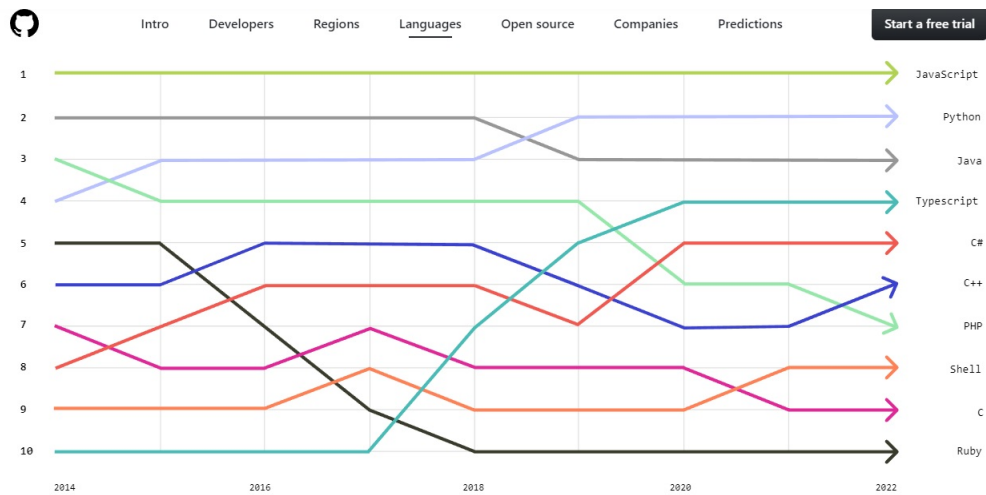


Figura A.2: Escala anual de Github sobre el uso de lenguajes de programación en los últimos años

ANEXO B

Diagramas de la Ingeniería del Software

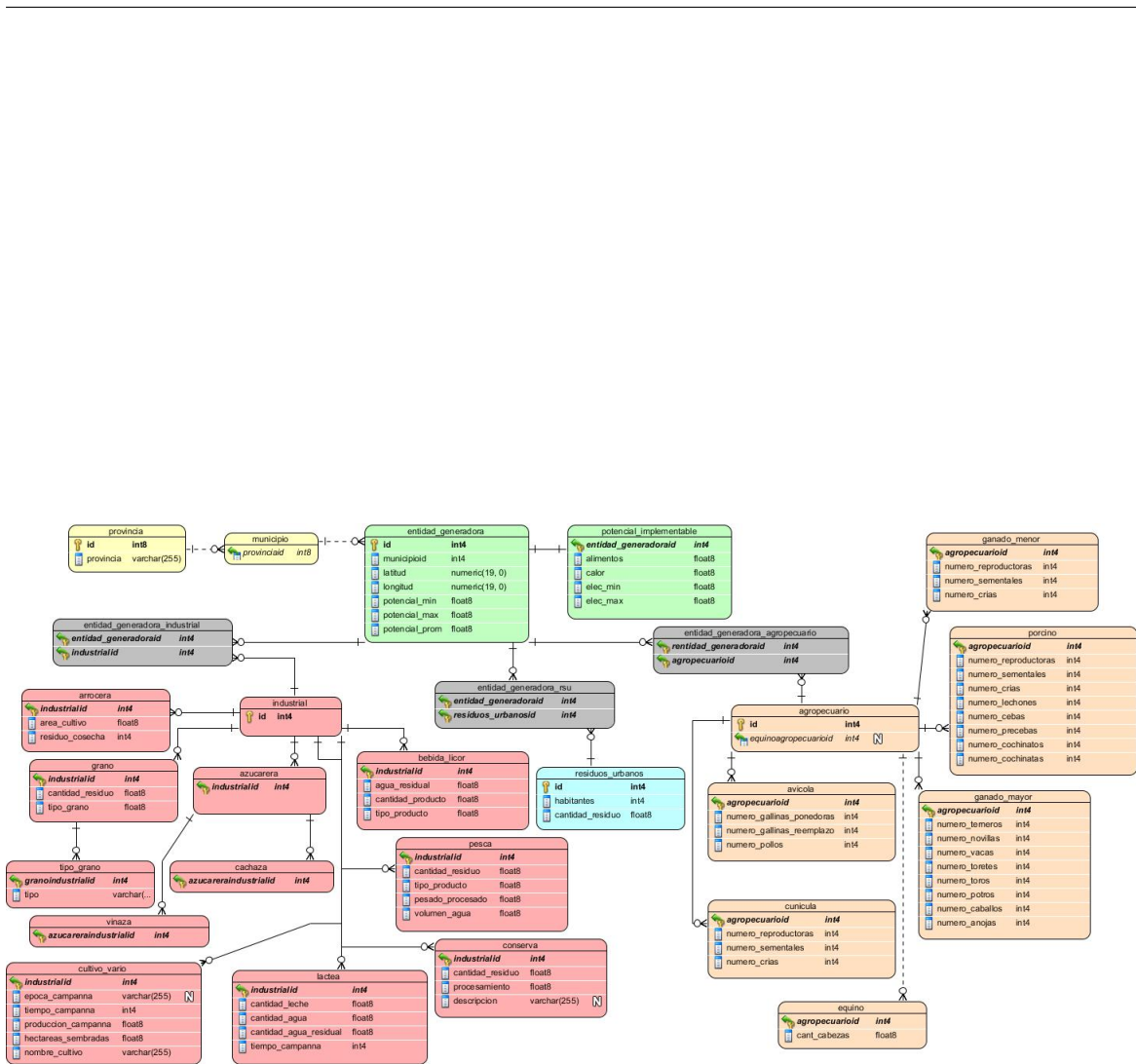


Figura B.1: Diagrama físico de la nueva base de datos

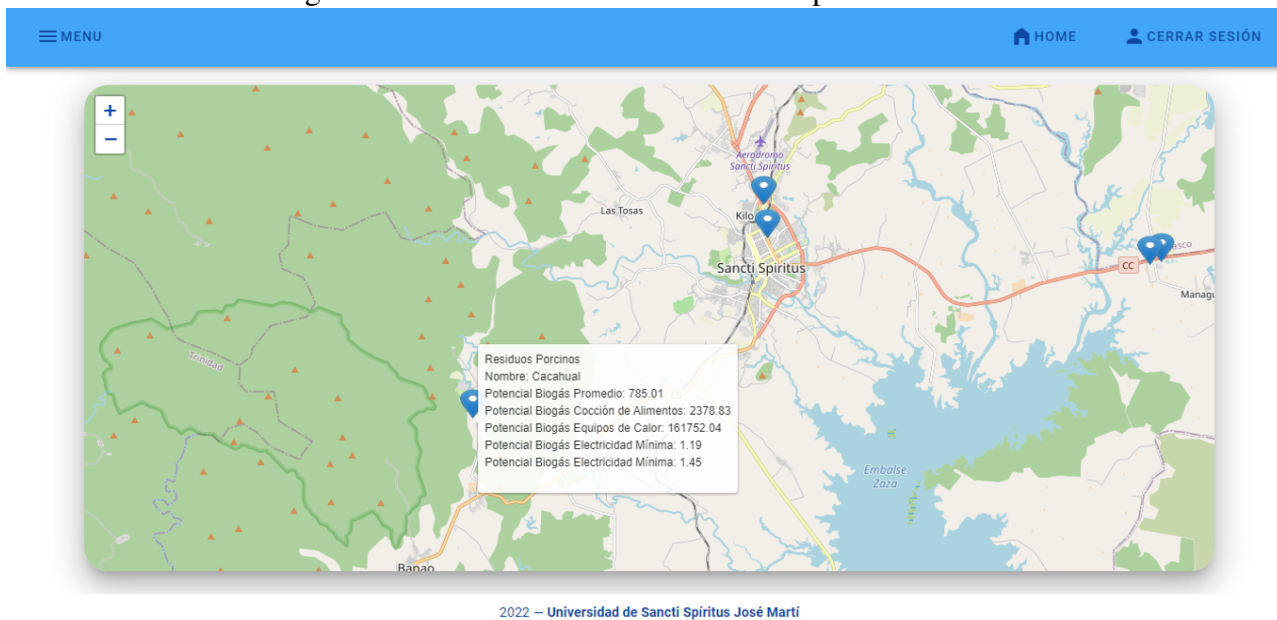
ANEXO C

Vistas de las aplicaciones

Figura C.1: Geolocalización de las aplicaciones antiguas



Figura C.2: Geolocalización de la nueva aplicación



ANEXO D

Manuales de Usuario

Frontend

Para la implementación del frontend en su computadora debe hacer lo siguiente:

Paso 1. Instalar node.js en la pc v16.14.2 e instalar git en la pc (descargar las instalaciones respectivamente)

Paso 2. Verificar la versión de node instalada abriendo el símbolo de sistema y ejecutando el comando: `node -v` 2.1 Instalar node version manager y node package manager con los comandos: `nvm install` y `npm install`

Paso 3. Crear una carpeta en cualquier lugar de la pc por ejemplo "nuevo-proyecto"

Ahora conectado a internet:

Paso 4. Abrir nuevamente el símbolo de sistema, yendo a dicha carpeta ejecutando los comandos: `cd "nuevo-proyecto" npm init`

Paso 5. Una vez creado el proyecto procedemos a instalar vuejs en la carpeta usando los comandos:

1. `npm init -y npm install vue`

2. `npm install -g @vue/cli npm init vue@latest (aquí es la versión 2 de vue) vue --version`

```
create-nuxt-app
Ok to proceed? (y) y
npm WARN deprecated source-map-url@0.4.1: See https://github.com/lydell/source-map-url#deprecated
npm WARN deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
npm WARN deprecated source-map-resolve@0.5.3: See https://github.com/lydell/source-map-resolve#deprecated
npm WARN deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated

create-nuxt-app v4.0.0
  Generating Nuxt.js project in nueva-app-nuxt
  Project name: nueva-app-nuxt
  Programming language: JavaScript
  Package manager: Npm
  UI framework: Vuetify.js
  Nuxt.js modules: Progressive Web App (PWA)
  Linting tools: Prettier
  Testing framework: None
  Rendering mode: Universal (SSR / SSG)
  Deployment target: Server (Node.js hosting)
  Development tools: (Press <space> to select, <a> to toggle all, <i> to invert selection)
  Continuous integration: None
  Version control system: Git
  Installing packages with npm
```

Figura D.1: Pasos en la consola para crear el proyecto nuxt

Paso 6. Ya verificada la versión de vuejs, procedemos a crear un proyecto nuxt con los comandos:

```
npm install -g
```

```
nuxt npm init nuxt-app nombre-del-proyecto
```

(la consola le irá preguntando lo siguiente, para lo que usted responderá:)

(esperar que termine la consola)

Paso 7 .Instalar dependencias de nuxt y vuejs que se utilizaron mediante el comando: `npm install nombre-de-la-dependencia --save` Estas son las dependencias instaladas tanto para el proceso de desarrollo como para el ambiente de producción:

1. @mdi/font
- 2.@nuxtjs/axios
3. @nuxtjs/color-mode
4. @nuxtjs/tailwindcss
5. axios 6.core-js
7. cors
8. export-from-json

9. leaflet

10. nuxt

11. nuxt-leaflet

12. vue

13. vue-json-excel

14. vue-router-user-roles

15. vue-select2

16. vue-server-renderer

17. vue-table-to-excel

18. vue-template-compiler

19. vue2-leaflet

20. vuetify

21. webpack

desarrollo : @nuxtjs/vuetify, vue2

Paso 8: Clonar el repositorio de frontend y copiarlo dentro de la carpeta creada de "nuevo-proyecto", sustituyendo los elementos

Paso 9. En la raíz de la carpeta "nuevo-proyecto" abrir el símbolo de sistema o la terminal deseada y ejecutar los comandos: npm run dev

Paso 10: Abrir el navegador en la dirección de localhost: 3000 o el puerto que diga la consola

Paso 11: Entrar al sistema como administrador o especialista Administrador es la contraseña Admin123*, la de especialista es Especialista123* .

Backend

1. Instalar postgresSQL

-
2. Crear una base de datos con el nombre "biogas"
 3. Dirigirse a github.com al repositorio de la universidad y clonar el repositorio del backend del proyecto de biogas 2022
 4. De vuelta al postgre: Hacer click derecho en la base de datos y restaurarla con el .backup descargado del repositorio del backend
 5. Dentro de la carpeta del repositorio clonado dirigirse a la carpeta de database config y poner allí la contraseña del postgresQL donde está la contraseña y el nombre de la base de datos biogas
 6. En la raíz de la carpeta del repositorio clonado ejecutar el comando: `npm run develop` 7. Abrir el navegador con la dirección "localhost: 1337" 8. Loguearse con su email y su contraseña

(de no funcionar los pasos mencionados anteriormente, crear una nueva base de datos vacía en postgresQL para ser usada luego: crear una carpeta nueva y, conectado a internet, ejecutar los comandos:

1. `npx create-strapi-app@latest nombre_del_proyecto`

(la consola va a preguntar lo siguiente, para lo que usted responderá:)

choose your installation type: Custom (manual settings)

choose your default database client: postgres

database name: (el nombre de la base de datos que creó para ser usada luego)

host: 127.0.0.1

port: 5432

username: postgres

password: (su contraseña del proceso de instalación de postgresql)

enable SSL connection: No (esperamos que la consola termine de trabajar)

ejecutar comandos que pida la consola: `cd nombre_del_proyecto npm run develop`

Parar el servidor con ctrl C y luego tomar la carpeta "api" dentro de "src" en el repositorio clonado y sustituirla por la carpeta "api" creada en este nuevo proyecto)