



ITSSAT

**INSTITUTO TECNOLÓGICO SUPERIOR
DE SAN ANDRÉS TUXTLA**



**Titulación Integral:
Tesis Profesional**

**Aplicación Web para la gestión de ganado
bovino para el rancho “El Mirador”**

**Que para obtener el título de:
Ingeniería Informática**

**Presenta:
Manuel Chipol Xolio**

**Asesor:
MTI. Rogelio Enrique Telona Torres**

San Andrés Tuxtla, 8 marzo de 2021

Formato de Liberación de Proyecto



FORMATO DE LIBERACIÓN DE PROYECTO PARA LA TITULACIÓN INTEGRAL

Lugar: San Andrés Tuxtla, Veracruz Fecha: 08/03/2021

Asunto: Liberación de proyecto para la Titulación integral.

LIC. GERMÁN VENTURA TENORIO
Jefe del Departamento de Estudios
Profesionales del ITSSAT

Por este medio informo que ha sido liberado el siguiente proyecto para la titulación integral:

Nombre del estudiante y/o egresado:	CHIPOL XOLIO MANUEL
Carrera:	INGENIERÍA INFORMÁTICA
No. de Control:	161U0570
Nombre del Proyecto:	SISTEMA PARA LA GESTION DE GANADO BOVINO PARA EL RANCHO EL MIRADOR
Producto:	1.-TESIS PROFESIONAL

Agradezco de antemano su valioso apoyo en esta importante actividad para la formación profesional de nuestros egresados.

ATENTAMENTE

L.I. GUADALUPE ZETINA CRUZ
JEFE DE DIVISIÓN DE INGENIERÍA INFORMÁTICA

 M.T.I. ROGELIO ENRIQUE TELONA TORRES	 M.T.I. JUAN RAFAEL GONZALEZ CADENA	 M.T.I. LORENZO DE JESUS ORGANISTA OLIVEROS
Nombre y firma del asesor	Nombre y firma del revisor*	Nombre y firma del revisor*

* Solo aplica para el caso de tesis o tesina.

Carr. Costera del Golfo S/N, KM 140+100
Loc. Matacapan, Mpio. San Andrés Tuxtla, Ver.
C.P. 95804 A.P. 110
Tel: 01(294)9479100 ext. 131
9479107

Autorización de Impresión



INSTITUTO TECNOLÓGICO SUPERIOR DE SAN ANDRÉS TUXTLA
DEPARTAMENTO DE ESTUDIOS PROFESIONALES



Autorización de Impresión para Titulación Integral

San Andrés Tuxtla, Ver. 08 de marzo del 2021

OPCION: 1.-TESIS PROFESIONAL

INTEGRANTES

Nombre del Estudiante	No. de Control	Carrera	Plan de Estudios
CHIPOL XOLIO MANUEL	161U0570	INGENIERÍA INFORMÁTICA	IINF-2010-220

De acuerdo con el Reglamento de Titulación Vigente y habiendo cumplido con todas las indicaciones que la Comisión Revisora le hizo a su trabajo profesional para la obtención del título a nivel licenciatura, comunico a Usted que este departamento concede la autorización para que proceda a la impresión del proyecto denominado:

"SISTEMA PARA LA GESTION DE GANADO BOVINO PARA EL RANCHO EL MIRADOR"

Atentamente


LIC. GERMAN VENTURA TENORIO
DEPARTAMENTO DE ESTUDIOS PROFESIONALES



Carr. Costera del Golfo S/N, KM 140+100
Loc. Matacapán, Mpio, San Andrés Tuxtla, Ver.
C.P. 95804 A.P. 110
Tel: 01(294) 94 79100 ext. 222
9479107

Agradecimientos

Dios tu mi única verdad y supremo, que ha permitido alcanzar mis objetivos en la vida, uno de ellos es terminar mi carrera de ingeniería informática.

A mis padres fuentes de inspiración que tuvieron esa confianza y me brindaron su apoyo económico de terminar mis estudios. Así como también a mi familia y amigos que de una u otra manera colaboran con la superación de lo antes mencionado.

A la escuela superior Instituto Tecnológico Superior de San Andrés Tuxtla, uno de los grandes institutos que me acepto y quien le debo ser profesional. A los docentes que nos impartieron conocimiento en las aulas de clases, formando un profesional con capacidad de servicio en el mundo laboral.

Índice de Contenido

Formato de Liberación de Proyecto	I
Autorización de Impresión	II
Agradecimientos.....	III
Índice de Contenido	IV
Lista de figuras.....	VIII
Lista de Tablas	XI
Resumen	XVII
Abstract	XVII
Introducción.....	1
CAPÍTULO I. GENERALIDADES	2
1.1 Descripción del Problema	2
1.2 Planteamiento del problema	2
1.3 Formulación del problema	3
1.4 Objetivos.....	4
1.4.1 Objetivo general.....	4
1.4.2 Objetivo específico	4
1.5 Justificación	5
CAPÍTULO II. FUNDAMENTOS TEÓRICOS	6
2.1 Antecedentes Teóricos.....	6
2.1.1 La ganadería y su importancia en el mundo	6
2.1.2 La ganadería en México.	7
2.2 Marco Teórico.....	8

2.2.1 Ingeniería de software	8
2.2.1.1 Definición.	8
2.2.1.2 Importancia	8
2.2.1.3 Técnicas de programación	10
2.2.2 Lenguajes de programación	11
2.2.2.1 Definición	11
2.2.2.2 Importancia	11
2.2.3 Lenguajes de programación más populares	13
2.2.3.1 Java Script	13
2.2.3.2 Java	14
2.2.3.3 PHP	14
2.2.3.4 SQL	14
2.3 Marco conceptual	15
CAPÍTULO III. METODOLOGÍA	16
3.1 Metodología SCRUM	16
3.2 Otras metodologías de desarrollo de software	17
3.2.1 Metodologías ágiles	17
3.2.2 Metodologías ágiles más utilizadas	17
3.2.3 Extreme Programming XP	17
3.2.4 Kanba	18
3.2.5 Agile Inception	18
3.2.6 Design Sprint, la metodología de google	19
3.3 Entorno de desarrollo	19
3.3.1 Editor Sublime Text 3	19
3.4 Tecnologías soportadas en el editor	20

3.5 Requisitos del sistema	20
3.5.1 Configuraciones mínimas de hardware	20
3.5.2 Configuraciones máximas de hardware	21
3.5.3 XAMPP	21
3.5.4 Licencia	22
3.6 Generalidades de gestores de base de datos	22
3.6.1 Definición	22
3.6.2 Importancia	22
3.6.3 Microsoft SQL Server	24
CAPÍTULO IV. Presentación, análisis e interpretación de resultados	25
4.1 Descripción de los resultados del trabajo	25
4.1.1 Creación de la base de datos	25
4.1.2 Inserción de registro	26
4.1.3 Consultas básicas	27
4.1.4 Creación de usuarios y privilegios.....	27
4.1.5 Programación de los esquemas de respaldo y recuperación	28
4.1.6 Programación de la aplicación	28
4.1.6.1 Index.php.....	28
4.1.6.2 Inicio.view.php.....	30
4.1.6.3 Ganados.view.php.....	32
4.1.6.4 Listadoganados.view.php.....	36
4.1.6.5 Procesarganado.php	39
4.1.6.6 Ganadoedit.view.php.....	41
4.1.6.7 Ganadodelete.php	45
4.1.6.8 Logout.php.....	46

4.1.6.9 Login_post.php	46
4.1.6.10 Functions.php	47
4.1.6.11 Connection.php	48
4.2 Resultados	49
4.2.1 Interfaces	49
4.3 Conclusiones	52
Recomendaciones.....	53
Referencia Bibliografía	55
Anexos	56

Lista de figuras

Diagrama de clase

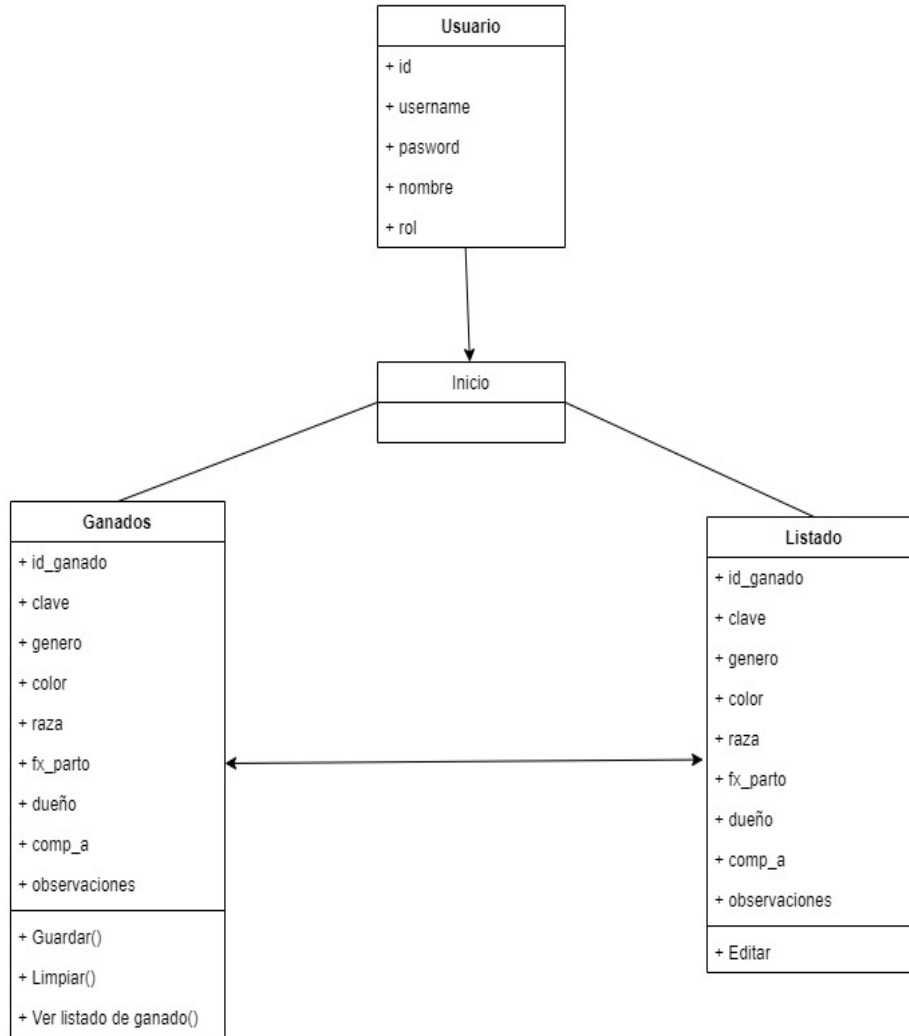


Diagrama de entidad relación

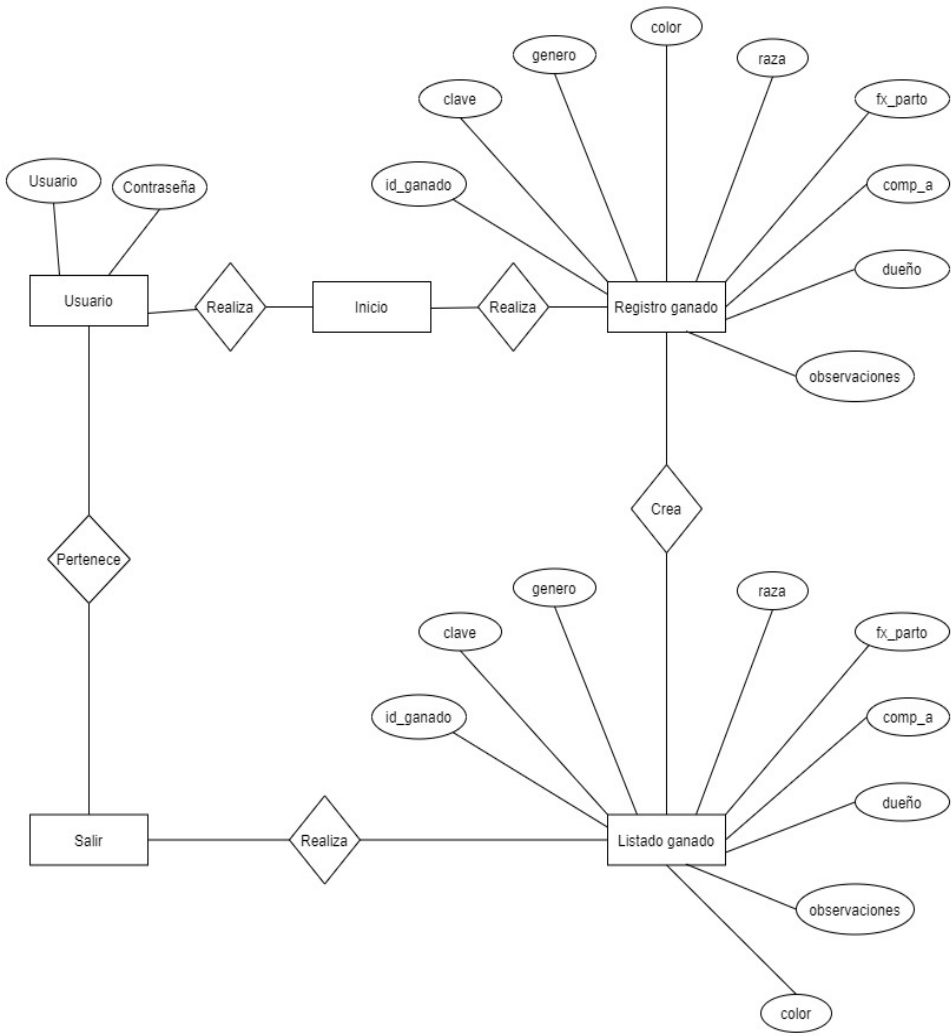


Diagrama de comunicación

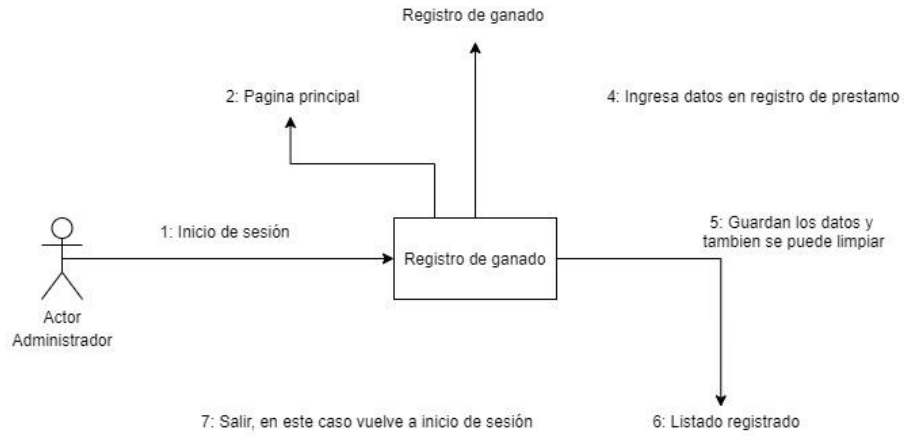
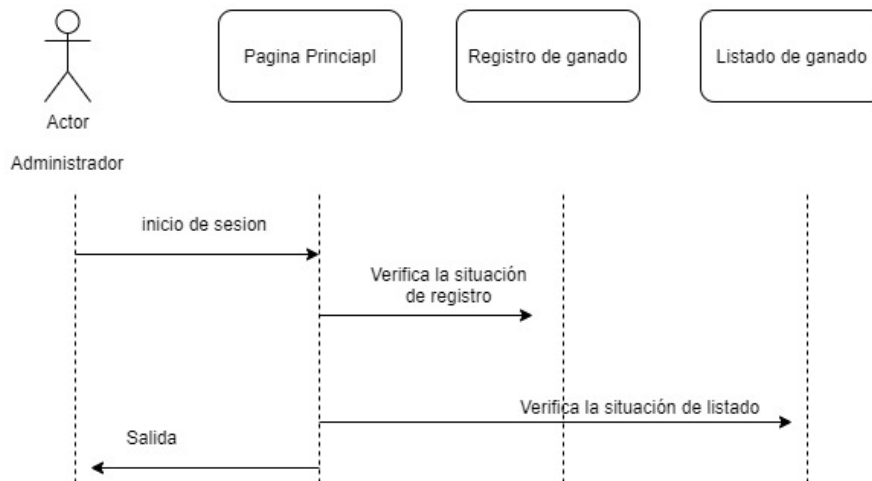


Diagrama de secuencia



Lista de Tablas

Lugar donde se realizará el proyecto.

Información sobre la empresa, institución u organización para la que se desarrollará el proyecto	
Nombre	El Mirador
RFC	CICM54010889A
Actividad	Ganadería
Localidad y Municipio	Península de Moreno, Mpio. de Catemaco

Elaborado por: Manuel Chipol Xolio

Caso de uso de inicio de sesión

# 1	CASODE USO: Inicio de Sesión
Actores: Administrador, Ganadero	Fecha: 15 de diciembre de 2020
Detalles: Caso de uso donde los usuarios acceden al sistema.	
Condiciones: <ul style="list-style-type: none"> • El usuario debe estar previamente creado en el sistema. • El usuario debe deberá tener un rol asignado y permisos de aplicación. 	
WORKFLOW: Flujo de Trabajo es la automatización de tareas, en este caso del sistema.	
Acción actor	Sistema
1.- Ingresa datos al usuario. 4.- Accede al sistema.	2.- Valida la información ingresada. 3.- Si la información es correcta muestra el menú principal de acuerdo a los permisos asignados previamente, de lo contrario regresa a la pantalla de LOGIN.
Fin: Usuario visualiza el menú y la interface del sistema.	

Elaborado por: Manuel Chipol Xolio

Caso de uso de registro de ganado

# 2	CASO DE USO: Registro de Ganado
Actores: Administrador, Ganadero	Fecha: 15 de diciembre de 2020
Detalles: Caso de uso donde se registran las cabezas de ganado.	
Condiciones: <ul style="list-style-type: none"> • El usuario debe de estar previamente creado en el sistema. • El usuario deberá tener un rol asignado y permisos de aplicación. 	
WORKFLOW	
Acción Actor	Sistema
1.- Ingresa información del ganado: <ul style="list-style-type: none"> • Ingresa ID. • Ingresa Clave. • Seleccionar Género en este caso sería: toro, vaca, novillo, novillona, becerro, becerra, ternero y ternera. • Ingresa Color. • Ingresar Raza. • Ingresar Fecha de Parto. • Ingresar Dueño. • Ingresar Comprado A: • Ingresar Observaciones • Guardar • Ver listado en caso de que no se allá registrado 	2.- Valida información ingresada. 3.- Si la información es correcta se crea el registro con la nueva cabeza de ganado.
Fin: Usuario registra una nueva cabeza de ganado bovino.	

Elaborado por: Manuel Chipol Xolio

Caso de uso de listado de ganado

# 3	CASO DE USO: Listado de Ganado
Actores: Administrador, Ganadero	Fecha: 15 de diciembre de 2020
Detalles: Caso de uso donde se ve el listado de las cabezas de ganado.	
Condiciones: <ul style="list-style-type: none"> • El usuario debe de estar previamente creado en el sistema. • El usuario deberá tener un rol asignado y permisos de aplicación. 	
WORKFLOW	
Acción Actor	Sistema
1.- Ver información del ganado: <ul style="list-style-type: none"> • ID. • Clave. • Género en este caso sería: toro, vaca, novillo, novillona, becerro, becerra, ternero y ternera. • Color. • Raza. • Fecha de Parto. • Dueño. • Comprado A: • Observaciones • Guardar • Ver listado en caso de que no se allá registrado. 	2.- Valida información ingresada. 3.- Si la información es correcta se ve el listado de las cabeza de ganado.
Fin: Usuario registra una nueva cabeza de ganado bovino o a inicio.	

Elaborado por: Manuel Chipol Xolio

Requerimientos funcionales

N°	Requisito	Detalle
# 1	Acceso al sistema (Pantalla de LOGIN)	Definir pantallas que validen la información del usuario de manera que garantiza la seguridad de la información.
# 2	Registro de Ganado	Crear una pantalla que permita el registro de los animales que existen o existirán dentro del sistema del ganadero, con sus respectivos detalles y sobre todo el aspecto más importante su factor identificador que en este caso sería la clave.
# 3	Listado de Ganado	Crear una pantalla que permita ver el listado de los animales que existen dentro del sistema del ganadero, con sus respectivos detalles y sobre todo el aspecto más importante su factor de identificador que es caso sería la clave.

Elaborado por: Manuel Chipol Xolio

Requerimientos no funcionales

N°	Requisito	Detalle
# 1	Alta Disponibilidad	El sistema estará disponible durante todos los días del año a excepción de mantenimientos correctivos y preventivos.
# 2	Multiplataforma	El sistema deberá poder ser utilizado bajo cualquier sistema operativo.
# 3	Usuarios	Definir usuarios para el uso del sistema propuesto.

Elaborado por: Manuel Chipol Xolio

Diccionario de la base de datos

Ganados			
Descripción: Registro de Ganado			
Descripción de los campos			
Campo	Tipo	Tamaño	Descripción
ID	Int	11	Cantidad de animales
Clave	Varchar	45	Código del animal
Genero	Text	45	Diferentes géneros
Color	Text	45	Colores de animales
Raza	Text	45	Razas de animales
Fecha de Parto	Timestamp	45	Fecha de nacimiento del animal
Dueño	Text	45	Propietario
Comprado a	Text	45	Emisor
Observaciones	Text	500	Observación de ingreso
Campo que se relaciona: (FK)		Campo que se relaciona: (PK)	

Fuente: Investigación directa
Elaborado por: Manuel Chipol Xolio

Ganados			
Descripción: Listado de Ganado			
Descripción de los campos			
Campo	Tipo	Tamaño	Descripción
ID	Int	11	Cantidad de animales
Clave	Varchar	45	Código del animal
Genero	Text	45	Diferentes géneros
Color	Text	45	Colores de animales
Raza	Text	45	Razas de animales
Fecha de Parto	Timestamp	45	Fecha de nacimiento del animal
Dueño	Text	45	Propietario
Comprado a	Text	45	Emisor
Observaciones	Text	500	Observación de ingreso
Campo que se relaciona: (FK)		Campo que se relaciona: (PK)	

Resumen

El objetivo principal de este proyecto fue crear un sistema informático que permita gestionar y controlar el registro de bovinos para el Rancho “El Mirador”, con el firme propósito de agilizar los procedimientos y de igual manera facilitar las actividades del día. La metodología a emplear es la conocida SCRUM ya que es un método de trabajo especialmente dirigido a proyectos en los que el cliente requiere de resultados rápidos, así mismo se considera que tiene tres fases una de ellas es la planificación la segunda la ejecución y la última optimización. Así mismo se elaboró el sistema con códigos como PHP, HTML, JAVA SCRIPT, CSS, MYSQL, de manera que se crearon carpetas como hojas de estilo, connection, functions, index, inicio, login_post, logout, ganadero, listadoganados, etc. Como servidor se ocupó XAMPP y para el tratamiento de fotos Photoshop, también para hacer los diagramas de entidad relación y de flujo, etc., se utilizó Draw.io como una herramienta para la elaboración de los diagramas.

Abstract

The main objective of this project was to create a computer system that allows the management and control of the cattle registry for the “El Mirador” Ranch, with the firm intention of streamlining the procedures and in the same way facilitating the activities of the day. The methodology to be used is the well-known SCRUM since it is a work method especially aimed at projects in which the client requires quick results, it is also considered that it has three phases, one of them is planning, the second is execution and the last optimization. Likewise, the system was developed with codes such as PHP, HTML, JAVA SCRIPT, CSS, MYSQL, so that folders were created such as style sheets, connection, functions, index, home, login_post, logout, rancher, cattle list, etc. . XAMPP server took care of and for Photoshop photo processing, also to make entity relationship and flow diagrams, etc., Draw.io was used as a tool for drawing diagram

Introducción

La ganadería representa un factor primario de mucha importancia para nuestro país, de manera que contribuye a la fertilidad de los campos, creando una gran riqueza para los parámetros productivos y reproductivos, sin embargo, se puede incrementar mediante cambios tecnológicos.

La ganadería “El mirador” es un rancho dedicado a la crianza y venta de ganado bovino, Ubicada en la comunidad Península de Moreno a 21 km de Municipio de Catemaco, Veracruz.

Se describe el desarrollo de la aplicación web que permita al administrador del rancho y ganadero guardar datos de su ganado, tales como síntomas, también permitirá registrar las fechas de vacuna, el tiempo de parto de una vaca, borrar un dato en la libreta, guarda la información. Actualmente este proceso se realiza de manera manual, ya que el ganadero tiene que llevar un control o registro de su ganado bovino, lo cual no permite garantizar una eficiencia y confianza al administrador, ya que pueda perder su registro o se le moje la libreta cuando ande en su trabajo.

CAPÍTULO I. GENERALIDADES

1.1 Descripción del Problema

Marco conceptual: Recolectar información acerca del tema de investigación en diferentes fuentes, siendo éstas electrónicas o documentales.

Determinación de requerimientos: Identificar los requisitos funcionales necesarios para el diseño del prototipo.

Diseñar prototipo: Elaborar el maquetado de la aplicación, que servirá como base al desarrollo del mismo.

Desarrollar prototipo: Realizar la aplicación mediante la tecnología seleccionada.

Prueba: Someter el prototipo a pruebas para determinar su adecuado funcionamiento.

1.2 Planteamiento del problema

Los ganaderos actualmente realizan todos sus procesos de manera manual, es decir registrando peso del bovino, fechas de vacuna, etc. en papel. Esto implica la posibilidad de no llevar un registro adecuado de datos importantes tanto para la administración como para un adecuado desarrollo del ganado. Estos son algunos de los síntomas, causas y consecuencias que sufre un ganadero. Derivado de lo anterior se plantea generar una aplicación web que dará solución al ganadero, brindando la confianza de guardar sus datos sin preocupación y para lo cual solo requerirá utilizar su celular con la aplicación web integrada.

Síntomas

- ✓ Que pierda el registro del ganado en una hoja común.
- ✓ Que olvide las fechas de vacuna.
- ✓ El tiempo de parto de una vaca.
- ✓ Borrar un dato en la libreta.
- ✓ Guarda la información.

Causas

Existen varias causas sobre la problemática de la ganadería ya que el ganadero tiene que llevar un control o registro de su ganado bovino, lo cual no permite garantizar eficiencia y confianza al administrador, ya que pueda perder su registro o se moje la libreta cuando ande en su trabajo.

Consecuencias

- ✓ Almacena datos de ganadería.
- ✓ Guardar la información con seguridad.
- ✓ Borra datos que no se requieren.
- ✓ Leer toda la información.
- ✓ Aplicar a la vida cotidiana del ganadero.
- ✓ Datos del registro de ganado bovino, equino, etc.

1.3 Formulación del problema

¿Desarrollar una aplicación web de control ganadero que permita guardar los datos?

Delimitación

El presente proyecto se desarrollará en el Rancho “El Mirador” de la comunidad Península de Moreno, perteneciente al Municipio de Catemaco, por lo que será diseñado a medida para utilizarlo en la gestión de ganado bovino, es importante mencionar que en esta primera fase solo se realizará el diseño y prueba del prototipo, pudiendo posteriormente robustecer la aplicación.

1.4 Objetivos

1.4.1 Objetivo general

Diseñar un prototipo de sistema web para la recopilación de datos del ganado vacuno para la administración del rancho “El Mirador”.

1.4.2 Objetivo específico

- Realizar el levantamiento de requerimientos funcionales y no funcionales del sistema.
- Diseñar el marco conceptual de la aplicación acorde con los requerimientos identificados.
- Desarrollar el prototipo basado en Tecnología Web.
- Someter a pruebas el prototipo desarrollado.

Cronograma

Actividad	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Marco conceptual	X	X	X	X											
Determinación de requerimientos				X	X	X									
Diseño de prototipo							X	X	X						
Desarrollar el prototipo										X	X	X	X		
Pruebas														X	X

1.5 Justificación

Los sistemas de identificación actualmente son de gran importancia ya que esta tecnología resulta de gran ayuda a la hora de llevar a cabo investigaciones, trabajos, innovaciones, creaciones y demás tareas. Gracias a ella se puede lograr un mayor entendimiento a cerca de lo estudiado en diferentes áreas del conocimiento toda vez que, de la mano de la tecnología, las diferentes áreas de estudio son capaces de mostrar nuevos descubrimientos o hallazgos que sirven de mucho a la sociedad.

Derivado de lo anterior, los sistemas de identificación de animales han permitido elevar el nivel de confianza por parte de los productores de ganado, incorporando nuevas tecnologías entre las cuales se destacan los sistemas de gestión que realizan su función de forma rápida y confiable que en otros tiempos eran inimaginables. Este sistema debe incluir la identificación, permitiendo controlar género, raza, etc. así como otros procesos derivados de la cría de estos animales.

En el rancho “El Mirador” Ubicada en la comunidad Península de Moreno a 21 km de Municipio de Catemaco, Veracruz; se realiza un seguimiento a cada animal desde el momento de su nacimiento o desde su compra hasta que deje de ser parte del rancho, ya sea por muerte o venta. Sin embargo, no cuenta con un sistema que permita agilizar las actividades que en él se realiza.

Por esta razón se propone desarrollar un prototipo web para el control administrativo, capaz de controlar ingresos y egresos de ganados, cambio de estatus, raza, etc. que permita al encargado organizar los datos de manera ordenada mediante la implementación de tecnología.

CAPÍTULO II. FUNDAMENTOS TEÓRICOS

2.1 Antecedentes Teóricos

2.1.1 La ganadería y su importancia en el mundo

Hoy en día la ganadería es la actividad agraria que consiste en la cría de animales la obtención de carne, leche o piletas, así mismo la carne es el valor económico para el ganadero. Sin embargo, uno de los mayores desafíos con los que se enfrenta en mundo es la manera de garantizar la seguridad alimentaria para la población urbana que se prevé que llegara a los 9 400 millones de habitantes para el año 2050, manteniendo al mismo tiempo su base de recursos naturales formada por el suelo, el agua, el aire y la diversidad biológica. La ganadería es y seguirá siendo un componente importante de la agricultura mundial.

La tecnología se ha vuelto muy importante en todas las actividades que realiza el hombre, ya que facilita sus tareas y le permite adquirir conocimientos. Un sistema de control ganadero le permite conocer y analizar de manera práctica toda una serie de indicadores individuales y poblaciones, esto constituye una excelente herramienta de gestión, selección y mejoramiento animal, con la que el negocio del ganadero, debe de mejorar e incrementar productividad y rentabilidad. Así mismo es importante mejorar el sistema de identificación de los animales para conocer su procedencia y saber la cantidad que hay.

La contribución de la ganadería a la economía mundial no se limita a la producción directa de alimentos, pues comprende también pieles, fibras, estiércol (fertilizante o combustible), fuerza de tiro y acumulación de capital. Estos elementos están estrechamente asociados con la estructura social y el bienestar de muchas sociedades rurales y sirven como reserva estratégica que aumenta la estabilidad no sólo del hogar, sino también del sistema agropecuario general.

Las exportaciones y la sustitución de las importaciones generan divisas y permiten

ahorrarlas. Tales contribuciones han estado en gran parte infravaloradas en el pasado y, aunque es evidente que va a disminuir la importancia relativa de estas funciones no alimentarias, si bien no de manera uniforme, es indudable que la ganadería seguirá satisfaciendo una amplia variedad de necesidades humanas.

2.1.2 La ganadería en México.

Como es del conocimiento público las actividades pecuarias se realizan en una amplia gama de sistemas productivos, que van desde los altamente tecnificados e integrados, hasta las de economías de tipo tradicional.

La actividad ganadera hace referencia a la domesticación de los animales como cerdos, vacas, pollos, borregos, etc., y es una actividad del sector primario que incluye su cuidado y alimentación, a su vez está enfocada básicamente en la generación de alimentos para consumo humano.

La Ganadería puede ser extensiva, intensiva y de autoconsumo, existen varios factores que influyen para un buen desarrollo de los animales como el relieve del suelo, acceso a fuentes de agua, un clima adecuado en cuanto a humedad y temperatura, así como la vegetación y el forraje que se utiliza para su alimentación. La ganadería fue introducida en México durante la época de la colonización, ya que en la época prehispánica se enfocaba de manera limitada a la cría de pavo, al xoloitzcuintle, a la cochinilla de grana y algunas especies apícolas.

Hoy en día el sector ganadero representa uno de los componentes con mayor crecimiento del sector agropecuario a nivel mundial, siendo así la carne de res, el segundo producto ganadero de mayor consumo, superado sólo por la de ave, principalmente pollo.

La producción de carne es la labor más diseminada en el medio rural, pues se realiza, sin excepción, en todas las zonas del país y aun en condiciones ambientales adversas que no permiten la práctica de otras actividades productivas. Existen tres tipos de centros de sacrificio, de inspección federal (TIF), municipales y privados. El sistema TIF minimiza el riesgo de que los productos y subproductos cárnicos puedan representar una fuente de zoonosis o diseminadores de enfermedades,

disminuyendo la afectación a la salud pública, la salud animal, la economía y el abasto nacional.

México es el séptimo productor mundial de proteína animal, esto es muestra del alto potencial de desarrollo que tiene y las ventajas competitivas que presenta el sector. SAGARPA a través de FIRCO otorga apoyos para el desarrollo de esta actividad con el Programa de Productividad y Competitividad Agroalimentaria para la instalación y uso de las energías renovables en los procesos productivos ganaderos, infraestructura y equipamiento para las diferentes instalaciones involucradas en el proceso de producción, instalación de cuartos fríos, por mencionar solo algunos.

2.2 Marco Teórico

2.2.1 Ingeniería de software

2.2.1.1 Definición.

La Ingeniería de software es la rama de la ingeniería que crea y mantiene las aplicaciones de software Ingeniería de Software aplicando tecnologías y prácticas de las ciencias computacionales, manejo de proyectos, ingeniería, el ámbito de la aplicación, y otros campos (Sandoval, s.f.).

2.2.1.2 Importancia

Una de las importancias de la Ingeniería de Software hace referencia a los actuales problemas que presenta su utilización y en muchos casos, su empleo totalmente empírico. Establece las principales situaciones como la incomprensión entre desarrolladores y usuarios o clientes, el largo tiempo de desarrollo, la corrección de fallas innecesariamente. Propone la enseñanza de la Ingeniería de Software aparejada con la de la programación como vía para eliminar la causa principal de lo planteado. Ejemplifica las ventajas del desarrollo de productos informáticos mediante el uso de una metodología de Ingeniería de Software con el Sistema de Negociación Comercial y el Sistema de Administración de las Capacidades. Hace

especial énfasis en cómo pueden desarrollarse de forma simultánea dos sistemas que tienen intercambio de información por medio de la utilización de la Ingeniería de Software (Ruíz y Aguilera 2007).

El manejo de sistemas informáticos es tan importante para los planes de las empresas, que en ocasiones se contratan gerentes con la única responsabilidad de administrarlos. Comprar la tecnología adecuada es la parte fácil; el reto es adaptarla a las necesidades de la organización y asimilarla; es por ello que alcanzar un alto grado de adaptación es un aspecto fundamental para el éxito de la compañía.

Es este precisamente el entorno donde se desarrolla la Ingeniería del Software, disciplina de la informática que integra proceso, métodos y herramientas para desarrollar y mantener software de calidad que resuelvan problemas de todo tipo. Ahora bien, la gestión de proyectos es una actividad importante dentro de la Ingeniería del Software. Empieza antes de iniciar cualquier actividad técnica y continúa a lo largo de todo el ciclo de vida del software. Comprende estimaciones, análisis de riesgos, planificación, seguimiento y control (Linares y Geizzelez, 2007).

Es importante medir el proceso de ingeniería de software y el producto que se elabora porque es la forma más objetiva de comprender y mejorar el proceso de desarrollo y el producto que se elabora. Si no se realizan mediciones, no hay forma de determinar si se está mejorando, las decisiones se basan sólo en evaluaciones subjetivas, lo que puede llevar a malas estimaciones o interpretaciones erróneas del proceso. Para establecer objetivos de mejora es necesario conocer el estado actual de desarrollo del software (Salazar, 2009).

2.2.1.3 Técnicas de programación

El diseño de un programa que se realiza sin seguir una metodología puede funcionar, pero se debe tener en cuenta que con el tiempo se convertirá en un conjunto de instrucciones. Es decir que las consecuencias de no utilizar un método determinado llevarán a cometer errores que pueden costar el buen funcionamiento del mismo. Las diferentes etapas del programa suelen tener discontinuidad y son difícilmente identificables. En consecuencia, existe una difícil fase de desarrollo y mantenimiento.

Aquí se identifican algunos de los problemas que suelen presentarse:

- Se presenta un exceso en la rigidez del programa, lo que implica que sean difícilmente adaptables a cualquier tipo de configuración.
- Se pierde mucho tiempo en la corrección de errores.
- Los programas generalmente son propios de cada programador, lo que implica que no sean muy manejables por otros.
- Para cuando se realice la documentación final, existirán deficiencias por presentarse la ausencia de diagramas, habrá descripciones que no estén completas o simplemente no las habrá, y la documentación no estará actualizada.

Existe una larga lista de problemas que pueden presentarse, pero en este caso sólo se han citado algunos. Es de suma importancia poder prevenir las modificaciones que puedan realizarse en el futuro, así como también la actualización de la documentación. Para esto, se citan algunas que son importantes como:

- Incrementar el volumen de datos y estructuras.
- Modificación en la forma como se organiza la información.
- Modificación por actualización de los documentos.
- Ampliación, reducción o sustitución en el sistema del proceso de datos.

La creación de programas debe tener la flexibilidad suficiente para ser modificables en el momento en que se requiera. Estos deben ser claros, simples, con el fin de poder ser leídos e interpretados de forma fácil, para la programación deberá asumirse ciertas normas que permitan la estandarización de la programación, implicando una disminución en costos, independencia del programador y seguridad.

Cuando existen problemas con cierto grado de complejidad, el diseño del algoritmo requiere de una reducción y simplificación en la legibilidad del algoritmo. Las técnicas de programación que sirven de gran ayuda para la solución de problemas de este tipo, consiguiendo mayor rapidez y eficacia tenemos:

- Estructurada
 - Secuencial
 - Repetitiva
 - Selección
- Modular

2.2.2 Lenguajes de programación

2.2.2.1 Definición

Un lenguaje de programación desde el punto de vista del usuario es una notación que sirve para escribir algoritmos que puedan ser ejecutados en una máquina; una notación precisa, rigurosa y con la pretensión de ausencia de ambigüedad. Un lenguaje tiene un conjunto de componentes elementales e indivisibles: variables, literales, palabras reservadas, símbolos separadores y símbolos aritméticos que pueden formar parte de un programa fuente.

2.2.2.2 Importancia

Un lenguaje de programación es un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y

expresiones. Es utilizado para controlar el comportamiento físico y lógico de una máquina. Aunque muchas veces se usan los términos 'lenguaje de programación' y 'lenguaje informático' como si fuesen sinónimos, no tiene por qué ser así, ya que los lenguajes informáticos engloban a los lenguajes de programación y a otros más, como, por ejemplo, el HTML (lenguaje para el marcado de páginas web que no es propiamente un lenguaje de programación).

Un lenguaje de programación permite a uno o más programadores especificar de manera precisa sobre qué datos debe operar una computadora, cómo estos datos deben ser almacenados o transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias. Todo esto, a través de un lenguaje que intenta estar relativamente próximo al lenguaje humano o natural, tal como sucede con el lenguaje Léxico.

Una característica relevante de los lenguajes de programación es precisamente que más de un programador puedan tener un conjunto común de instrucciones que puedan ser comprendidas entre ellos para realizar la construcción del programa de forma colaborativa. Los procesadores usados en las computadoras son capaces de entender y actuar según lo indican programas escritos en un lenguaje fijo llamado lenguaje de máquina. Todo programa escrito en otro lenguaje puede ser ejecutado de dos maneras:

- Mediante un programa que va adaptando las instrucciones conforme son encontradas. A este proceso se le llama interpretar y a los programas que lo hacen se los conoce como intérpretes.
- Traduciendo este programa, al programa equivalente escrito en lenguaje de máquina. A ese proceso se le llama compilar y al programa traductor se le denomina compilador (Balderrama, s.f.).

López, (s.f.) considera que el paradigma de programación ha evolucionado, y entre muchos avances que se han producido, destacan tres grandes pasos: el primero es

cuando se inventó la programación de computadoras como tal; el segundo paso es cuando se gestó la primera gran evolución, a la que se le nombró Programación Estructurada; y el tercer paso, es cuando se generó otra gran evolución, a la que se le denomina Programación Orientada a Objetos.

2.2.3 Lenguajes de programación más populares

- C
- C++
- C#
- VISUAL BASIC.NET
- HTML
- MYSQL
- PYTHON
- CSS

2.2.3.1 Java Script

No debemos confundirlo con Java. Son lenguajes distintos. Javascript es un lenguaje de programación que puede ser utilizado para crear programas que luego son acoplados a una página web o dentro de programas más grandes. Podemos ver funcionando este lenguaje en servicios de chat, calculadoras o buscadores de información.

2.2.3.2 Java

Reconocido por su legibilidad y simplicidad, Java es uno de los lenguajes de programación más adoptados: más 9 millones de desarrolladores lo usan y está presente en 7 mil millones de dispositivos en todo el mundo. Desde 2001 se mantiene en las primeras posiciones.

Su enorme popularidad se debe a su poder de permanencia, cuestión que asegura el funcionamiento a largo plazo de las aplicaciones que lo utilizan.

2.2.3.3 PHP

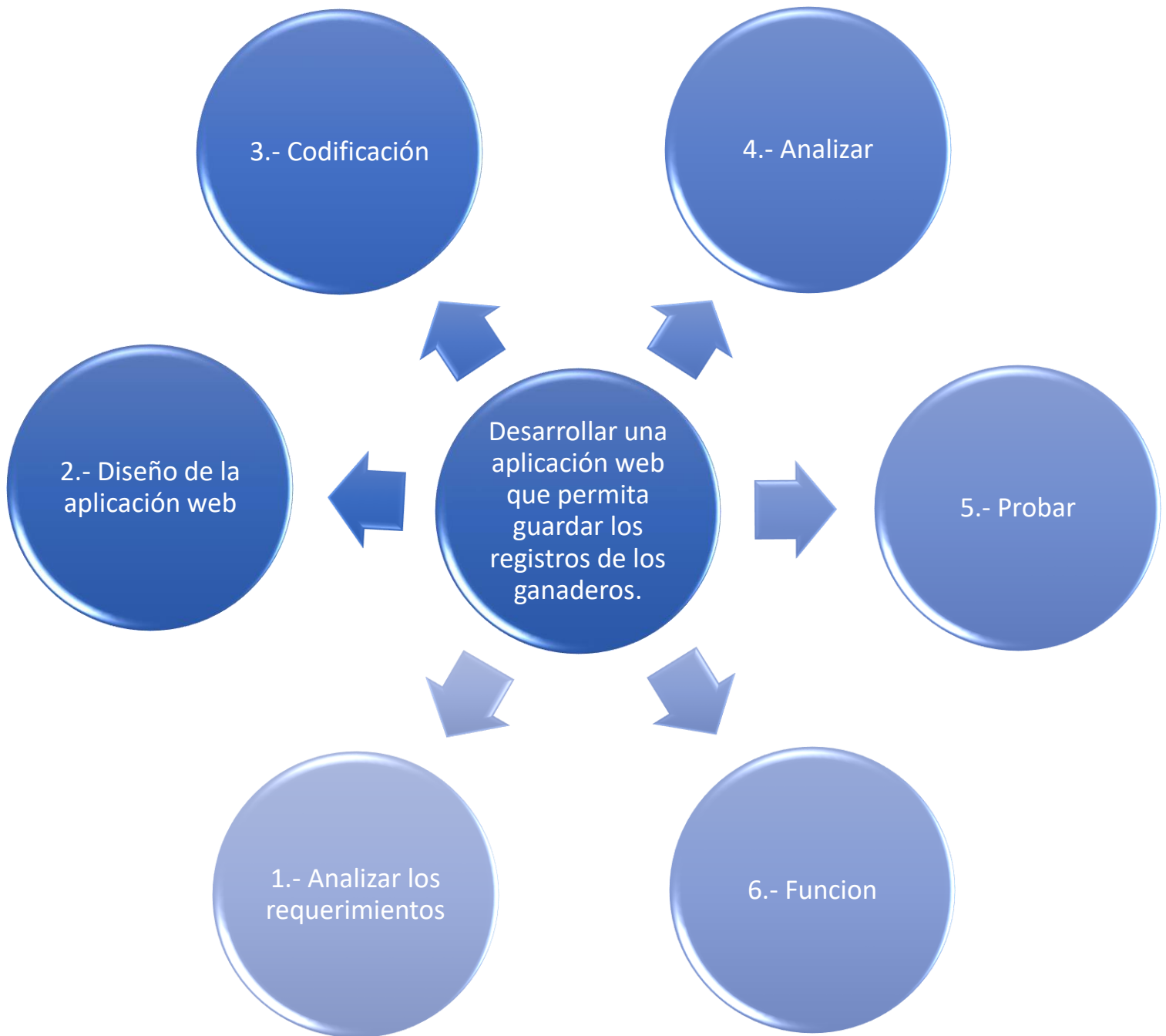
Creado en 1994 por el programador canadiense Rasmus Lerdorf, con la intención de contar con un conjunto de herramientas para el mantenimiento de las páginas web y no como lenguaje.

Es de fácil acceso para nuevos programadores y a su vez ofrece grandes herramientas a los más experimentados.

2.2.3.4 SQL

Este lenguaje de programación ha sido diseñado para administrar, proteger y recuperar los datos de sistemas de gestión de información, lo cual ha sido utilizado fuertemente en los últimos años a partir del desarrollo de la ciberseguridad

2.3 Marco conceptual



CAPÍTULO III. METODOLOGÍA

3.1 Metodología SCRUM

La metodología a emplear en este proyecto es la conocida como SCRUM, es un método de trabajo especialmente dirigido a proyectos en los que el cliente requiere de resultados rápidos (entregas parciales y regulares) en un entorno de cambios constantes, es decir, donde la capacidad de reacción debe ser muy alta.

Con la metodología Scrum se consigue la entrega de un desarrollo web o de una aplicación en un entorno flexible y colaborativo y en el tiempo y costes planeados.

El framework Scrum se basa en ciclos cortos y regulares de trabajo (llamados iteraciones) en los que debe completarse una serie de tareas y entregarse un resultado concreto a partir de unos objetivos y requisitos marcados previamente (backlog product) para cada uno de los ciclos. Resumimos brevemente las fases de la metodología Scrum:

Planificación: planificar la iteración, seleccionando los requisitos prioritarios y definiendo las tareas. Cada fase se considerará un 'sprint' –normalmente de un mes de duración como máximo- en el que se presentará el producto evolucionado al cliente para ser testeado.

Ejecución: se ejecuta la primera iteración con reuniones diarias de sincronización (daily meetings), para analizar y valorar el desarrollo de las tareas asignadas a cada miembro del equipo según el Sprint Planning.

Optimización: antes de que finalice el ciclo se realiza una reunión de revisión de la iteración con el cliente, en el que se presenta la evolución del producto, realizándose los cambios oportunos en base al desarrollo realizado. Asimismo, se analiza el trabajo realizado, las dificultades y avances para optimizar el proceso de trabajo.

Entrega final del producto: tras completar todos los ciclos de iteraciones.

3.2 Otras metodologías de desarrollo de software

Una metodología es una colección de procedimientos, desarrollo, técnica, documentos auxiliares y herramientas que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información.

3.2.1 Metodologías ágiles

Las metodologías ágiles son aquellas que permiten adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo flexibilidad e inmediatez en la respuesta para amoldar el proyecto y su desarrollo a las circunstancias específicas del entorno.

En esencia, las empresas que apuestan por esta metodología consiguen gestionar sus proyectos de forma flexible, autónoma y eficaz reduciendo los costes e incrementando su productividad.

3.2.2 Metodologías ágiles más utilizadas

Actualmente existen diferentes opciones, pero las más utilizadas son: programación extrema (XP), Scrum y Kanban, todas ellas se guían a través de un patrón establecido por el Manifiesto Ágil realizado por varios autores que establecieron los 12 principios del software ágil.

3.2.3 Extreme Programming XP

Esta herramienta es muy útil sobre todo para startups o empresas que están en proceso de consolidación, puesto que su principal objetivo es ayudar en las relaciones entre los empleados y clientes. La clave del éxito del Extreme Programming XP es potenciar las relaciones personales, a través, del trabajo en equipo, fomentando la comunicación y eliminando los tiempos muertos.

Sus principales fases son:

- Planificación del proyecto con el cliente.
- Diseño del proyecto.
- Codificación, donde los programadores trabajan en pareja para obtener resultados más eficientes y de calidad.
- Pruebas para comprobar que funcionan los códigos que se van implementando

3.2.4 Kanba

La estrategia Kanban conocida como Tarjeta Visual muy útil para los responsables de proyectos. Esta consiste en la elaboración de un cuadro o diagrama en el que se reflejan tres columnas de tareas; pendientes, en proceso o terminadas. Este cuadro debe estar al alcance de todos los miembros del equipo, evitando así la repetición de tareas o la posibilidad de que se olvide alguna de ellas. Por tanto, ayuda a mejorar la productividad y eficiencia del equipo de trabajo.

Las ventajas que proporciona esta metodología son:

- Planificación de tareas.
- Mejora en el rendimiento de trabajo del equipo.
- Métricas visuales.
- Los plazos de entregas son continuos

3.2.5 Agile Inception

Está orientada a la definición de los objetivos generales de las empresas. Su meta es clarificar cuestiones como el tipo de cliente objetivo, las propuestas de valor añadido, las formas de venta. Suele girar en torno al método de elevator pitch, que

consiste en pequeñas reuniones entre los socios y el equipo de trabajo en las que las intervenciones no pueden superar los 5 minutos.

3.2.6 Design Sprint, la metodología de google

En cualquier organización, la estrategia de negocios es lo más importante. Las metodologías ágiles se llevan implementando desde hace una década con el fin de mejorar los procesos que llevan a un producto o servicio mejorado y de calidad en el que los clientes cobran cada vez más importancia. Como ejemplo de innovación en estrategias de negocios nos encontramos con Design Sprint, una metodología de Google que está favoreciendo a los perfiles profesionales del mundo agile.

Esta metodología viene de la mano de Google Ventures, un servicio del gigante tecnológico para la innovación y promoción de startups tecnológicas. Se trata de un proceso que dura 5 días en el que el negocio tiene que resolver todas las cuestiones relacionadas con diseño, prototipado, testeo de clientes. La idea es que el trabajo se elabora en etapas de sprints en las que meses de trabajo se pueden reducir en pocas semanas, en vez de esperar a lanzar un producto para entender si la idea es buena, el prototipo proporciona antes la información para evitar posibles errores.

3.3 Entorno de desarrollo

El presente proyecto se desarrolló con el IDE de Sublime Text 3, lo cual tiene lo necesario para desarrollar el sistema web de control ganadero del Rancho “El Mirador”.

3.3.1 Editor Sublime Text 3

Los editores de código son muy buenos aliados a la hora de trabajar en desarrollo Web. Sublime Text 3 es una herramienta que nos brinda varias opciones muy interesantes para trabajar en diversas plataformas, ya que cuenta con versiones para Microsoft Windows, Mac OS X y Linux.

La primera versión fue publicada inicialmente en el año 2008. La versión 2 vio la luz

en el año 2011 en release Alpha. El lanzamiento definitivo de Sublime Text 2 fue anunciado en el 2012 y con ella, este software logró mayor solidez y robustez, mejoras en la interfaz y también el soporte a las pantallas Retina de MacBook Pro. ¿Algunas ventajas para elegir este editor de código? Minimapa de navegación para desplazarnos con comodidad por toda la extensión del código, selección múltiple (multiselección), diferentes configuraciones de layout, Plugin API, autocompletado y coloreado inteligente de código.

3.4 Tecnologías soportadas en el editor

El sistema de resaltado de sintaxis de Sublime Text soporta un gran número de lenguajes así como: (C, C++, C#, CSS, D, Erlang, HTML, Groovy, Haskell, HTML, Java, JavaScript, LaTeX, Lisp, Lua, Markdown, Matlab, OCaml, Perl, PHP, Python, R, Ruby, SQL, TCL, Textile and XML).

3.5 Requisitos del sistema

Una herramienta que nos brinda varias opciones muy interesantes para trabajar en diversas plataformas, ya que cuenta con versiones para Microsoft Windows, Mac OS X y Linux.

3.5.1 Configuraciones mínimas de hardware

Microsoft Windows Vista SP1 / Windows 7 Professional:

- Procesador: 800MHz Intel Pentium III o equivalente.
- Memoria: 512 MB.
- Espacio en disco: 750 MB de espacio libre en disco

Ubuntu 9.10:

- Procesador: 800MHz Intel Pentium III o equivalente.
- Memoria: 512 MB.
- Espacio en disco: 650 MB de espacio libre en disco

Macintosh OS X 10.7 Intel.

- Procesador: Intel de doble núcleo.
- Memoria: 2 GB.
- Espacio en disco: 650 MB de espacio libre en disco

3.5.2 Configuraciones máximas de hardware

Microsoft Windows 7 Professional / Windows 8 / Windows 10:

- Procesador: Intel Core i9 o equivalente.
- Memoria: 8 GB (32 bits), 16 GB (64 bits).
- Espacio en disco: 1 TB de espacio libre en disco.

Ubuntu 15.04:

- Procesador: Intel Core i5 o equivalente.
- Memoria: 2 GB (32 bits), 4 GB (64 bits).
- Espacio en disco: 1.5 GB de espacio libre en disco

OS X 10.10 Intel:

- Procesador: Intel de doble núcleo.
- Memoria: 4 GB.
- Espacio en disco: 1.5 GB de espacio libre en disco

3.5.3 XAMPP

Es un entorno más popular del desarrollo con PHP, también es distribución de apache completamente gratuita y fácil de instalar que contiene MariaDB, PHP y Perl. El paquete de instalación de XAMPP ha sido diseñado para ser increíblemente fácil de instalar y usar.

Esa aplicación está disponible para las siguientes plataformas:

Windows

Linux

OS X

3.5.4 Licencia

Es una compilación de software libre (similar a una distribución de Windows). Es gratuita y puede ser copiada libremente de acuerdo a la licencia GNU GPL. Únicamente la compilación de xampp esta publicada bajo la licencia GPL. Cada uno de los componentes incluidos tiene su propia licencia y debería consultarla para conocer que es posible y que no. en el caso del uso comercial deberás consultar las licencias individuales, en particular MYSQL. Desde el punto de vista de xampp como compilación del uso comercial es gratuito.

3.6 Generalidades de gestores de base de datos

3.6.1 Definición

Los sistemas de Gestión de Bases de Datos, son aplicaciones que permiten a los usuarios definir, crear y mantener la base de datos y proporciona un acceso controlado a la misma.

3.6.2 Importancia

Tener conocimiento de la evolución real de los datos en un sistema se requiere tener información sobre el tiempo. Ejemplo de tales aplicaciones son la administración de oficinas, asuntos bancarios, registro de personal, registros médicos, administración de inventarios, monitoreo del tiempo, etc. Las bases de datos tradicionales carecen de capacidades para registrar y manejar las variaciones dependientes del tiempo del mundo que modelan. Es por ello que han aparecido diferentes soluciones para dotar a los Sistemas de gestión de bases de datos (SGBD) de posibilidades de operar con el tiempo. Una base de datos temporal soporta algún aspecto de tiempo. Una base de datos relacional es temporal si contiene una o más relaciones temporales. Las bases de datos temporales permiten el almacenamiento y

recuperación de todos los estados asumidos por un objeto durante su tiempo de vida, y con ello registran su evolución en el tiempo. Se caracterizan por la incorporación de uno o más atributos temporales, que pueden reflejar el momento en que un hecho fue actualizado en la base de datos (tiempo de transacción), o el momento en el que realmente ocurrieron los hechos que se modelan (tiempo válido), o ambos inclusive. En dependencia de la información temporal con que se trabaje, estas bases de datos se clasifican como Bases de datos tradicionales (Snapshot), Bases de datos de tiempo de transacción (BDTT), Bases de datos de tiempo válido (BDTV) y Bases de datos bitemporales (BDBT) (Martínez y Díaz, 2010). Control sobre la redundancia de datos: Los sistemas de archivos almacenan varias copias de los mismos datos en archivos distintos. Esto hace que se desperdicie espacio de almacenamiento, además de provocar la falta de consistencia de datos. En los sistemas de bases de datos todos estos archivos están integrados, por lo que no se almacenan varias copias de los mismos datos. Sin embargo, en una base de datos no se puede eliminar la redundancia completamente, ya que en ocasiones es necesaria para modelar las relaciones entre los datos, o bien es necesaria para mejorar las prestaciones.

Los SGBD es la aplicación que interactúa con los usuarios de los programas de aplicación y la base de datos. Algunos de los SGBD más conocidos son:

- DB2
- SLQ/DS
- ORACLE
- INGRES
- INFORMIXSQL

Entre los objetivos de un sistema de gestión de base de datos (SGBD) se detallan:

- Definir la Base de Datos mediante el Lenguaje de Definición de Datos, el cual permite especificar la estructura, tipo de datos y las restricciones sobre los datos, almacenándolo todo en la base de datos.
- Separar la descripción y manipulación de la data, permitiendo un mayor entendimiento de los objetos, además de flexibilidad de consulta y actualización de los datos:

- Permitir la inserción, eliminación, actualización, consulta de los datos mediante el Lenguaje de Manejo de Datos, lo que permite resolver el problema que presentan los sistemas de archivos, donde hay que trabajar con un conjunto fijo de consultas o la necesidad de tener muchos programas de aplicaciones.

3.6.3 Microsoft SQL Server

Microsoft SQL Server. Es un sistema para la gestión de bases de datos creado por Microsoft, el mismo se basa en el modelo relacional. SQL Server utiliza como lenguajes de consulta T-SQL y ANSI SQL.

Microsoft SQL Server revoluciona el concepto de Base de datos para la Empresa. Reúne en un sólo producto la potencia necesaria para cualquier aplicación empresarial, crítica junto con unas herramientas de gestión que reducen al mínimo el coste de propiedad. Con Microsoft SQL Server, la empresa tiene todo de serie.

Dentro de sus características fundamentales se encuentran:

- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información.
- Además, permite administrar información de otros servidores de datos.

CAPÍTULO IV. Presentación, análisis e interpretación de resultados

4.1 Descripción de los resultados del trabajo

4.1.1 Creación de la base de datos

```
CREATE DATABASE IF NOT EXISTS `CGC`;  
USE `CGC`;
```

```
DROP TABLE IF EXISTS `ganados`;
```

```
CREATE TABLE `ganados` (  
  `id_ganado` int(11) NOT NULL AUTO_INCREMENT,  
  `clave` varchar(45) NOT NULL,  
  `genero` text(45) NOT NULL,  
  `color` text(45) NOT NULL,  
  `raza` text(45) NOT NULL,  
  `fx_parto` timestamp(6) NOT NULL,  
  `dueño` text(45) NOT NULL,  
  `comp_a` text(45) NOT NULL,  
  `observaciones` text(500) NOT NULL,  
  PRIMARY KEY (`id_ganado`)  
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `listado` (  
  `id_ganado` int(11) NOT NULL AUTO_INCREMENT,  
  `clave` varchar(45) NOT NULL,  
  `genero` text(45) NOT NULL,  
  `color` text(45) NOT NULL,
```

```

`raza` text(45) NOT NULL,
`fx_parto` timestamp(6) NOT NULL,
`dueño` text(45) NOT NULL,
`comp_a` text(45) NOT NULL,
`observaciones` text(500) NOT NULL,
PRIMARY KEY (`id_ganado`),
KEY `FK_ganados_idganados_idg` (`id_ganado`),
CONSTRAINT `FK_ganados_idganados_idg` FOREIGN KEY (`id_ganado`)
REFERENCES `ganados` (`id_ganado`) ON DELETE NO ACTION ON UPDATE
NO ACTION,
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8;

```

4.1.2 Inserción de registro

```
LOCK TABLES `ganados` ganados;
```

```

INSERT INTO ganados VALUES (1,00306824-6460,'Vaca','Negro','Suizo',8-01-
2021,'Manuel', 'Ninguno', 'Ninguno'),
(2,00306824-6461,'Toro','Sardo','Sardo Negro',8-02-2021,'Manuel', 'Ninguno',
'Ninguno');

```

```
UNLOCK TABLES;
```

Inserción de registro para el usuario:

```

INSERT INTO `users` VALUES (1,'admin','admin123','Manuel','Administrador'),(2,
'ganadero', 'ganadero123', 'Manuel', 'Ganadero'), (3, 'Enrique', 'telona123',
'Telona','Profesor');

```

4.1.3 Consultas básicas

USE biblioteca;

SELECT * From ganados;

SELECT titulo FROM listado;

4.1.4 Creación de usuarios y privilegios

Para crear usuarios se coloca el siguiente comando:

```
CREATE TABLE `users` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `username` varchar(15) NOT NULL,  
  `password` varchar(15) NOT NULL,  
  `nombre` varchar(60) NOT NULL,  
  `rol` varchar(15) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8;
```

El siguiente comando nos permite conceder permisos para poder acceder y usar el servidor MySQL:

```
GRANT ALL PRIVILEGES ON *.* TO 'nombre del usuario'@'localhost, ip , o %  
remotamente' IDENTIFIED BY 'contraseña';
```

Para guardar los datos ingresamos el siguiente comando: FLUSH PRIVILEGES;

Para eliminar a los usuarios ingresamos el siguiente comando; DROP USER 'nombre del usuario'@'localhost, ip o %';

4.1.5 Programación de los esquemas de respaldo y recuperación

Para la crear una copia de seguridad de la base de datos se escribe la siguiente línea de comando: `mysqldump -u root -p (nombre la base) > (ubicación)(nombre.sql)` Así mismo para restaurar una base de datos desde el respaldo con extensión `.sql` se escribe la siguiente línea de comando: `mysqldump -u root -p (nombre de la base) < (ubicación) (nombre.sql)`

4.1.6 Programación de la aplicación

4.1.6.1 Index.php

```
<?php
//arreglo con mensajes que puede recibir
$messages = [
    "1" => "Credenciales incorrectas",
    "2" => "No ha iniciado sesión"
];
?>
<!DOCTYPE html>
<html>
<head>
<title>Login | Registro de Ganado</title>
    <meta name="description" content="Control Ganadero - Rancho El Mirador" />
    <link rel="stylesheet" href="css/style.css" />
    <link rel="Shortcut Icon" type="image/x-icon" background= "imagen/inicio9.jpg"
/>

</head>
<body align="left" class="full-cover-background" width="100" height="100"
style="background-image:url(imagen/fondo1.jpg);">
<div class="header">
```

```

<div class="form-container" >
  <div class="transparente" >
    <br>
    <br>

    <h1>
      <marquee behavior=alternate >Control Ganadero - Rancho "El
Mirador</marquee></h1>

  </div>
  <br>
  <br>
  <br>
  <br>
  <br>
  <br>
  <div class="body">
    <div class="panel-login">
      <center></center>
      <h4 class="text-center all-tittles" style="margin-bottom: 40px;">Inicio de
Sesión</h4>
      <form method="post" class="form" action="login_post.php">
        <label>Usuario: </label><br>
        <input type="text" name="username">
        <br>
        <label>Contraseña: </label><br>
        <input type="password" name="password">
        <br><br>
        <center><button type="submit">Entrar</button></center>

```

```

        </form>
    <?php
        if(isset($_GET['err']) && is_numeric($_GET['err']) && $_GET['err'] > 0 &&
$_GET['err'] < 3 )
            echo '<span class="error">'.$messages[$_GET['err']].'</span>';
        ?>
    </div>
</div>

<footer>
    <p>Creado &copy; Manuel Chipol Xolio 2021</p>

</footer>

</body>

</html>

```

4.1.6.2 Inicio.view.php

```

<!DOCTYPE html>
<?php
require 'functions.php';
$permisos = ['Administrador','Ganadero','Profesor'];
permisos($permisos);

?>
<html>
<head>
<title>Inicio | Registro de Ganado</title>

```



```

<meta name="description" content="Control Ganadero - Rancho El Mirador" />
<link rel="stylesheet" href="css/style.css" />
<link rel="Shortcut Icon" type="image/x-icon" href="imagen/logo.jpg" />

</head>
<body background="imagen/vaca.jpg" width="0" height="0">
</center>
<div class="header" align="center">
    
    <div align="center"> <h3>Usuario: <?php echo $_SESSION["username"]
?></h3></div>
</div>

<nav>
    <ul>
        <li class="active"><a href="inicio.view.php">Inicio</a> </li>
        <li><a href="ganados.view.php">Registro de Ganado</a> </li>
        <li><a href="listadoganados.view.php">Listado de Ganado</a> </li>
        <li class="right"><a href="logout.php">Salir</a> </li>

    </ul>
</nav>

<div class="body">
    <div class="panel">
        <h1 class="text-center">Control Ganadero - Rancho "El Mirador"</h1>
        <?php
        if(isset($_GET['err'])){
            echo '<h3 class="error text-center">ERROR: Usuario no autorizado</h3>';

```

```
}
?>
<br>
<hr>
<p class="text-center"><strong>Ubicación</strong>
    <br><br>Península de Moreno, Mpio. Catemaco, Ver.<br>Rancho "El
Mirador"<br>Tel: 2941405932<br>manuel.chipol@gmail.com</p>
<br>
</div>
</div>

<footer>

    <p>2021 &copy; Manuel Chipol Xolio</p>
</footer>

</body>

</html>
```

4.1.6.3 Ganados.view.php

```
<!DOCTYPE html>
<?php
require 'functions.php';
//Define quienes tienen permiso en este archivo
$permisos = ['Administrador','Ganadero','Profesor'];
permisos($permisos);

?>
```

```

<html>
<head>
<title>Inicio | Registro de Ganado</title>
  <meta name="description" content="Control Ganadero - Rancho El Mirador" />
  <link rel="stylesheet" href="css/style.css" />
  <link rel="Shortcut Icon" type="image/x-icon" href="imagen/logo.jpg" />

</head>
<body>
<div class="header" align="center">
  <h1>Control Ganadero - Rancho "El Mirador"</h1>
  <h3>Usuario: <?php echo $_SESSION["username"] ?></h3>
</div>
<nav>
  <ul>
    <li><a href="inicio.view.php">Inicio</a> </li>
    <li class="active"><a href="ganados.view.php">Registro de Ganado</a> </li>
    <li><a href="listadoganados.view.php">Listado de Ganado</a> </li>
    <li class="right"><a href="logout.php">Salir</a> </li>

  </ul>
</nav>

<div class="body">
  <div class="panel">
    <h4>Registro de Ganado</h4>
    <form method="post" class="form" action="procesarganado.php">

```

```
<br><br>
<label>ID</label><br>
<input type="number" min="1" class="number" name="id">
<br><br>
<label>Clave</label><br>
<input type="text" required name="clave" maxlength="45">
<br><br>
<label>Genero</label><br>
<input required type="radio" name="genero" value="Toro"> Toro
<input type="radio" name="genero" required value="Vaca"> Vaca
<input type="radio" name="genero" required value="Novillo"> Novillo
<input type="radio" name="genero" required value="Novillona"> Novillona
<input type="radio" name="genero" required value="Becerro"> Becerro
<input type="radio" name="genero" required value="Becerra"> Becerra
<input type="radio" name="genero" required value="Ternero"> Ternero
<input type="radio" name="genero" required value="Ternera"> Ternera

<br><br>
<label>Color</label><br>
<input type="text" required name="color" maxlength="45">

<label>Raza</label><br>
<input type="text" required name="raza" maxlength="45">

<label>Fecha_Part0</label><br>
<input type="text" required name="fecha_parto" maxlength="45">

<label>Dueño</label><br>
<input type="text" required name="dueño" maxlength="45">
```

```
<label>Comprado a</label><br>
<input type="text" required name="comprado_a" maxlength="45">

<label>Observaciones</label><br>
<input type="text" required name="observaciones" maxlength="45"><br><br>
<button type="submit" name="insertar">Guardar</button> <button
type="reset">Limpiar</button> <a class="btn-link" href="listadoganados.view.php">Ver
Listado de Ganado</a>
```

```
<br><br>
<!--mostrando los mensajes que recibe a traves de los parametros en la url-->
<?php
if(isset($_GET['err']))
    echo '<span class="error">Error al almacenar el registro</span>';
if(isset($_GET['info']))
    echo '<span class="success">Registro almacenado correctamente!</span>';
?>
```

```
</form>
<?php
if(isset($_GET['err']))
    echo '<span class="error">Error al guardar</span>';
?>
</div>
</div>
```

```
<footer>
<p>2021 &copy; Manuel Chipol Xolio</p>
```

```
</footer>
```

```
</body>
```

```
</html>
```

4.1.6.4 Listadoganados.view.php

```
<?php
```

```
require 'functions.php';
```

```
$permisos = ['Administrador','Ganadero','Profesor'];
```

```
permisos($permisos);
```

```
// tengo una duda--consulta los ganados para el listado de ganados configurar
```

```
$ganados = $conn->prepare("select a.id_ganado, a.clave, a.genero, a.color,
```

```
a.raza, a.fx_parto, a.dueño, a.comp_a, a.observaciones");
```

```
$ganados->execute();
```

```
$ganados = $ganados->fetchAll();
```

```
?>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Listado de ganado | Registro de Ganado</title>
```

```
<meta name="description" content="Control Ganadero - Rancho El Mirador" />
```

```
<link rel="stylesheet" href="css/style.css" />
```

```
<link rel="Shortcut Icon" type="image/x-icon" href="imagen/logo.jpg" />
```

```
</head>
```

```

<body>
<div class="header" align="center">
    <h1>Control Ganadero - Rancho "El Mirador"</h1>
    <h3>Usuario: <?php echo $_SESSION["username"] ?></h3>
</div>
<nav>
    <ul>
        <li><a href="inicio.view.php">Inicio</a> </li>
        <li><a href="ganados.view.php">Registro de Ganado</a> </li>
        <li class="active"><a href="listadoganados.view.php">Listado de Ganado</a>
</li>
        <li class="right"><a href="logout.php">Salir</a> </li>

    </ul>
</nav>

<div class="body">
    <div class="panel">
        <h4>Listado de Ganado</h4>
        <table class="table" cellspacing="0" cellpadding="0">
            <tr>
                <th>ID<br></th><th>Clave</th><th>Genero</th><th>Color</th><th>Raza</th><th>
                >Fecha de Parto</th><th>Dueño</th><th>Comprado
                <th>Observaciones</th><th>Editar</th><th>Eliminar</th>
            </tr>
            <tr>
                <td align="center"><?php echo $ganado['id_ganado'] ?></td>
                <td><?php echo $ganado['clave'] ?></td>
                <td><?php echo $ganado['genero'] ?></td>

```

```

        <td><?php echo $ganado['color'] ?></td>
        <td><?php echo $ganado['raza'] ?></td>
        <td align="center"><?php echo $ganado['fx_parto'] ?></td>
        <td><?php echo $ganado['dueño'] ?></td>
        <td><?php echo $ganado['comp_a'] ?></td>
        <td><?php echo $ganado['observaciones'] ?></td>

        <td><a href="ganadoedit.view.php?id_ganado=<?php echo
$ganado['id_ganado'] ?>">Editar</a> </td>
        <td><a href="ganadodelete.php?id_ganado=<?php echo
$ganado['id_ganado'] ?>">Eliminar</a> </td>
    </tr>
    <?php endforeach;?>
</table>
<br><br>

<a class="btn-link" href="ganados.view.php">Agregar Ganado</a>
<br><br>
<!--mostrando los mensajes que recibe a traves de los parametros en la
url-->
<?php
if(isset($_GET['err']))
    echo '<span class="error">Error al almacenar el registro</span>';
if(isset($_GET['info']))
    echo '<span class="success">Registro almacenado
correctamente!</span>';
?>

</div>
</div>

```



```
<footer>
  <p>2021&copy; Manuel Chipol Xolio</p>
</footer>

</body>

</html>
```

4.1.6.5 Procesarganado.php

```
<?php
if(!$_POST){
  header('location: ganados.view.php');
}
else {
  //incluimos el archivo funciones que tiene la conexion
  require 'functions.php';
  //Recuperamos los valores que vamos a llenar en la BD
  $id = htmlentities($_POST ['id_ganado']);
  $clave = htmlentities($_POST ['clave']);
  $genero = htmlentities($_POST['genero']);
  $color = htmlentities($_POST['color']);
  $raza = htmlentities($_POST['grado']);
  $fecha_parto = htmlentities($_POST['fx_parto']);
  $dueño = htmlentities($_POST ['dueño']);
  $comprado_a = htmlentities($_POST ['comp_a']);
  $observaciones = htmlentities($_POST ['observaciones']);

  //insertar es el nombre del boton guardar que esta en el archivo
  ganados.view.php
```

```

if (isset($_POST['insertar'])){

    $result = $conn->query("insert into ganados (id_ganado, clave, genero, color,
raza, fx_parto, dueño, comp_a, observaciones) values ('$id', '$clave', '$genero',
'$color', '$raza', '$fecha_parto', '$dueño', '$comprado_a', '$observaciones' )");
    if (isset($result)) {
        header('location:ganados.view.php?info=1');
    } else {
        header('location:ganados.view.php?err=1');
    }
    // validación de registro

//sino boton modificar que esta en el archivo ganadoedit.view.php
} else if (isset($_POST['modificar'])) {
    //capturamos el id ganados a modificar
    $id_ganado = htmlentities($_POST['id_ganado']);
    $result = $conn->query("update ganados set id_ganado = '$id', clave =
'$clave', genero = '$genero', color = '$color', raza = '$raza', fx_parto =
'$fecha_parto, dueño = '$dueño', comp_a = '$comprado_a, observaciones =
'$observaciones' where id = " . $id_ganado);
    if (isset($result)) {
        header('location:ganadoedit.view.php?id=' . $id_ganado . '&info=1');
    } else {
        header('location:ganadoedit.view.php?id=' . $id_ganado . '&err=1');
    }
    // validación de registro
}
}
}

```

4.1.6.6 Ganadoedit.view.php

```
<!DOCTYPE html>
<?php
require 'functions.php';
$permisos = ['Administrador','Ganadero','Profesor'];
permisos($permisos);
if(isset($_GET['id_ganado'])) {

    $id_ganado = $_GET['id'];

    $ganado = $conn->prepare("select * from ganados where id_ganado =
".$id_ganado);
    $ganado->execute();
    $ganado = $ganado->fetch();

}else{
    Die('Ha ocurrido un error');
}
?>
<html>
<head>
<title>Inicio | Registro de Ganado</title>
    <meta name="description" content="Control Ganadero - Rancho El Mirador" />
    <link rel="stylesheet" href="css/style.css" />
    <link rel="Shortcut Icon" type="image/x-icon" href="imagen/institution.png" />

</head>
<body>
<div class="header">
    <h1>Control Ganadero - Rancho "El Mirador"</h1>
    <h3>Usuario: <?php echo $_SESSION["username"] ?></h3>

```

```

</div>
<nav>
  <ul>
    <li><a href="inicio.view.php">Inicio</a> </li>
    <li class="active"><a href="ganados.view.php">Registro de Ganado</a> </li>
    <li><a href="listadoganados.view.php">Listado de Ganado</a> </li>
    <li class="right"><a href="logout.php">Salir</a> </li>

  </ul>
</nav>

<div class="body">
  <div class="panel">
    <h4>Edición de Ganado</h4>
    <form method="post" class="form" action="procesarganado.php">
      <!--colocamos un campo oculto que tiene el id del alumno-->
      <input type="hidden" value="<?php echo $ganado['id']?>" name="id">

      <label>ID</label><br>
      <input type="number" min="1" class="number" value="<?php echo
$ganado['id_ganado']?>" name="id">
      <br><br>
      <label>Clave</label><br>
      <input type="text" required name="clave" value="<?php echo
$ganado['clave']?>" maxlength="45">
      <br><br>
      <label>Genero</label><br>
      <input required type="radio" name="genero" <?php if($ganado['genero']
== 'Toro'){ echo "checked";} ?> value="Toro"> Toro
      <input required type="radio" name="genero" <?php if($ganado['genero']

```

```

== 'Vaca'){ echo "checked";} ?> value="Vaca"> Vaca
    <input required type="radio" name="genero" <?php if($ganado['genero']
== 'Novillo'){ echo "checked";} ?> value="Novillo"> Novillo
    <input required type="radio" name="genero" <?php
if($ganado['genero'] == 'Novillona'){ echo "checked";} ?> value="Novillona">
Novillona
    <input required type="radio" name="genero" <?php
if($ganado['genero'] == 'Becerro'){ echo "checked";} ?> value="Becerro"> Becerro
    <input required type="radio" name="genero" <?php
if($ganado['genero'] == 'Becerra'){ echo "checked";} ?> value="Becerra"> Becerra
    <input required type="radio" name="genero" <?php
if($ganado['genero'] == 'Ternero'){ echo "checked";} ?> value="Ternero"> Ternero
    <input type="radio" name="genero" required value="Ternera" <?php
if($ganado['genero'] == 'Ternera') { echo "checked";} ?>> Ternera
    <br><br>
    <label>Color</label><br>
    <input type="text" required name="color" value="<?php echo
$ganado['color']?>" maxlength="45">
    <br><br>
    <label>Raza</label><br>
    <input type="text" required name="raza" value="<?php echo
$ganado['raza']?>" maxlength="45">
    <br><br>
    <label>Fecha_Partido</label><br>
    <input type="text" required name="fecha_partido" value="<?php echo
$ganado['fx_partido']?>" maxlength="45">
    <br><br>
    <label>Dueño</label><br>
    <input type="text" required name="dueño" value="<?php echo
$ganado['dueño']?>" maxlength="45">
    <br><br>

```

```

        <label>Comprado_a</label><br>
        <input type="text" required name="comprado_a" value="<?php echo
$ganado['comp_a']?>" maxlength="45">
        <br><br>
        <label>Observaciones</label><br>
        <input type="text" required name="observaciones" value="<?php echo
$ganado['observaciones']?>" maxlength="45">
        <br><br>

        <button type="submit" name="modificar">Guardar Cambios</button> <a
class="btn-link" href="listadoganados.view.php">Ver Listado de Ganado</a>
        <br><br>
        <!--mostrando los mensajes que recibe a traves de los parametros en la
url-->

        <?php
        if(isset($_GET['err']))
            echo '<span class="error">Error al editar el registro</span>';
        if(isset($_GET['info']))
            echo '<span class="success">Registro modificado
correctamente!</span>';
        ?>

        </form>
    </div>
</div>

<footer>
    <p>Creado &copy; Manuel Chipol Xolio 2021</p>
</footer>

```

```
</body>
```

```
</html>
```

4.1.6.7 Ganadodelete.php

```
<?php
require 'functions.php';
if($_SESSION['rol'] == 'Administrador') {
    if (isset($_GET['id_ganado']) && is_numeric($_GET['id_ganado'])) {
        try {
            $id_ganado = $_GET['id_ganado'];
            $ganado = $conn->prepare("delete from ganados where id_ganado = " .
            $id_ganado);
            $ganado->execute();
            header('location:listadoganados.view.php');
        } catch (PDOException $e) {
            echo $e->getMessage();
        }
    } else {
        die('Ha ocurrido un error');
    }
} else {
    header('location:inicio.view.php?err=1');
}
?>
```

4.1.6.8 Logout.php

```
<?php
session_start();
session_destroy();
header('location:index.php');
?>
```

4.1.6.9 Login_post.php

```
<?php
if ($_SERVER['REQUEST_METHOD'] == 'POST') {

    require 'conn/connection.php';

    $username = htmlspecialchars($_POST['username']);
    $password = htmlspecialchars($_POST['password']);

    $users = $conn->prepare("select username, password, rol from users where
username = '". $username. "' and password = '". $password. "'");
    $users->execute();
    if($users->rowCount() > 0){
        $user = $users->fetch();
        if($user['username'] == $username && $user['password'] == $password){
            //existe el usuario y esa contraseña
            session_start();
            $_SESSION["username"] = $username;
            $_SESSION["rol"] = $user['rol'];
            setcookie("activo", 1, time() + 3600);
            header("Location: inicio.view.php", true, 301);
        } else {
            http_response_code(401);
        }
    }
}
```



```

        //echo "Credenciales incorrectas";
        header('location:index.php?err=1');
    }
} else {
    http_response_code(401);
    header('location:index.php?err=1');
}
} else {
    http_response_code(405);
    echo "SOLO SE PUEDE POST";

    // POST_GET
}

```

4.1.6.10 Functions.php

```

<?php
//iniciamos la sesion
session_start();

//esta pregunta la debe hacer en todos los archivos para validar que antes el
usuario haya iniciado sesion
if ( isset($_COOKIE["activo"]) && isset($_SESSION['username'])) {
    setcookie("activo", 1, time() + 3600);
} else {
    http_response_code(403);
    header('location:index.php?err=2');
}

//importamos el archivo que contiene la variable de conexion a la base de datos
require 'conn/connection.php';

//para verificar que tiene acceso a un archivo

```

```

function permisos($permisos){
    if (!in_array($_SESSION['rol'], $permisos)) {
        http_response_code(403);
        header('location:inicio.view.php?err=1');
    }
}

//eliminar
function ganados($id_ganado, $id_materia, $conn){
    $ganados = $conn->prepare("select * from ganados where id_ganado =
'$id_ganado' and id_ganado = '$id_ganado'");
    $ganados->execute();
    //si devuelve una fila significa que la nota ya es
    $ganados = $ganados->rowCount();
    return $ganados;
}

?>

```

4.1.6.11 Connection.php

```

<?php
try{
    $conn = new PDO('mysql:host=localhost; dbname=CGC', 'root', '');
} catch(PDOException $e){
    echo "Error: ". $e->getMessage();
    die();
}
?>

```

4.2 Resultados

En este punto se anexan capturas de pantalla del proyecto elaborado para el rancho “El Mirador”.

4.2.1 Interfaces

Como se puede observar en este pantalleo se muestra el inicio de sesión para el rancho el “El Mirador”, de manera que pide el usuario y contraseña tal y como como se muestra en la fig. 4.2.1.

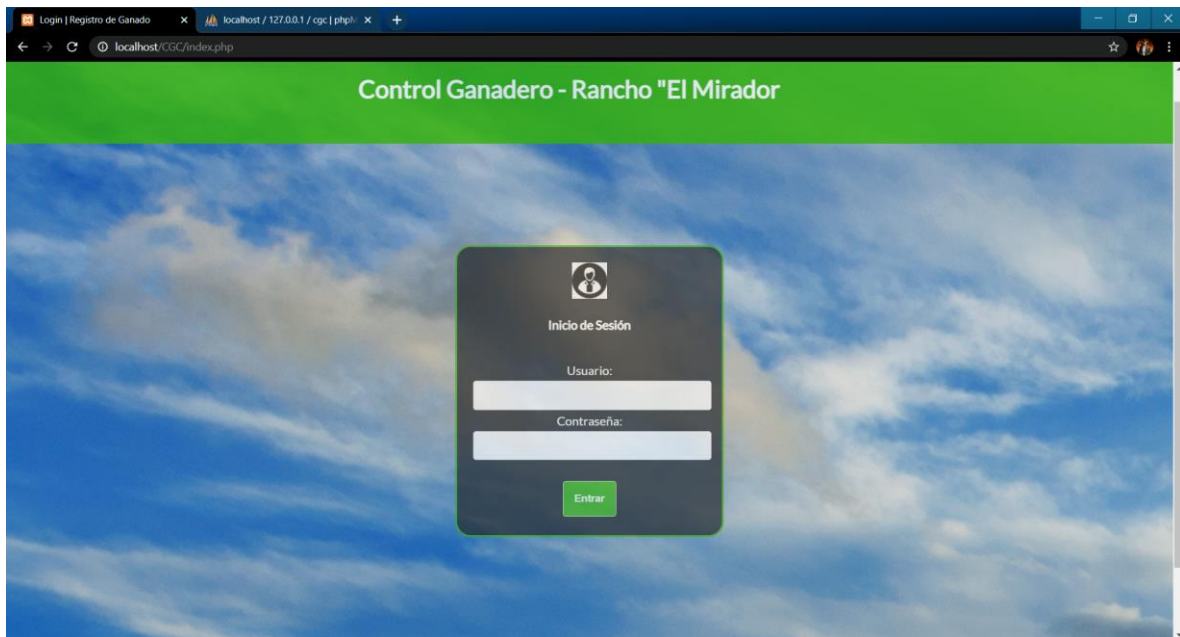


Fig. 4.2.1 Ventana de inicio de sesión

Una vez que hayamos ingresado, nos mostrará la página principal del sistema control ganadero Rancho “El mirador” de manera que ahí se encuentra su ubicación y todos los datos del rancho. Fig. 4.2.2

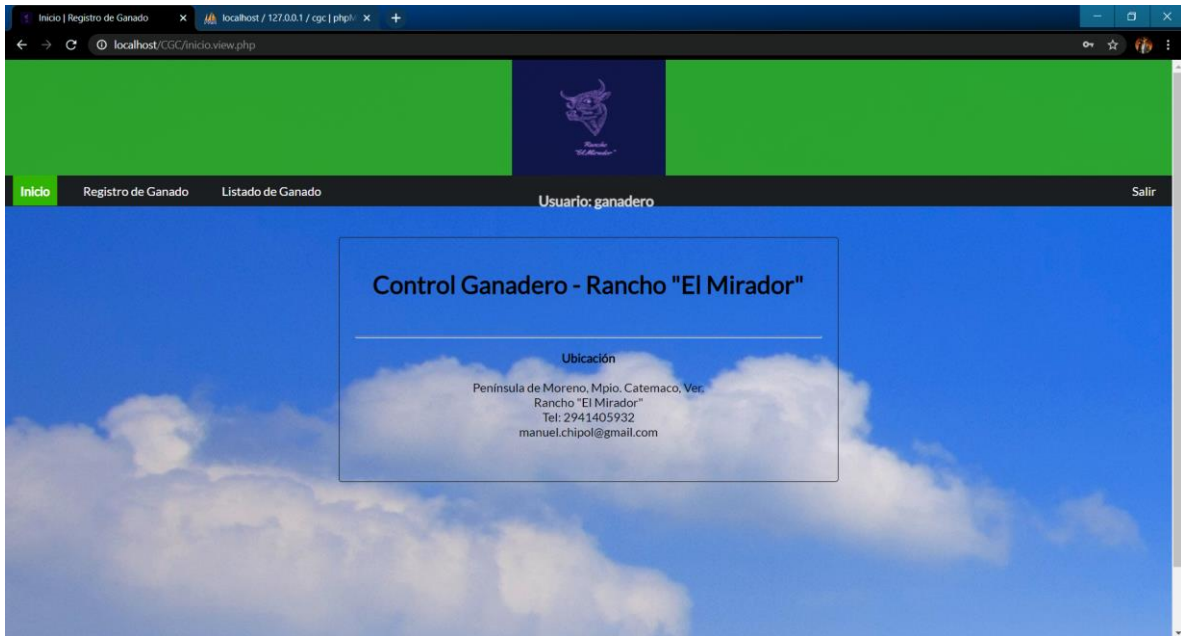


Fig. 4.2.2 Ventana de inicio del sistema

En esta ventana se hacen los registros de ganados, pide el id, clave, genero, color, raza, fecha de parto, dueño, comprado a y observaciones. Esta sección, también permite guardar, limpiar y ver el listado, tal y como se muestra en la figura 4.2.3

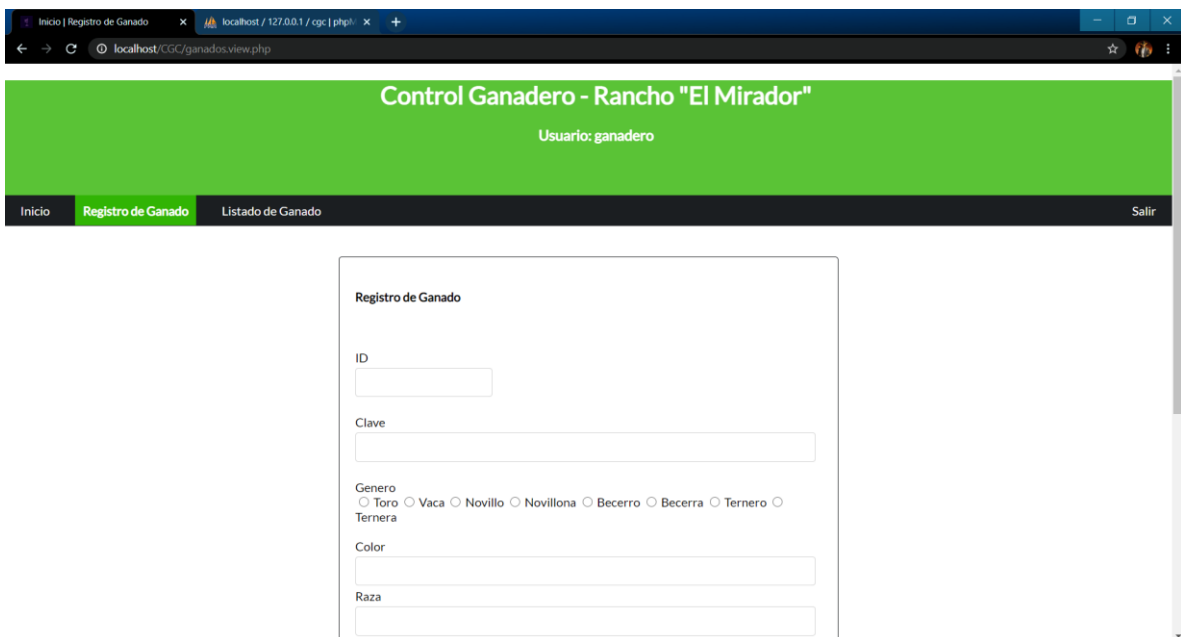


Fig. 4.2.3 Ventana registro de ganado

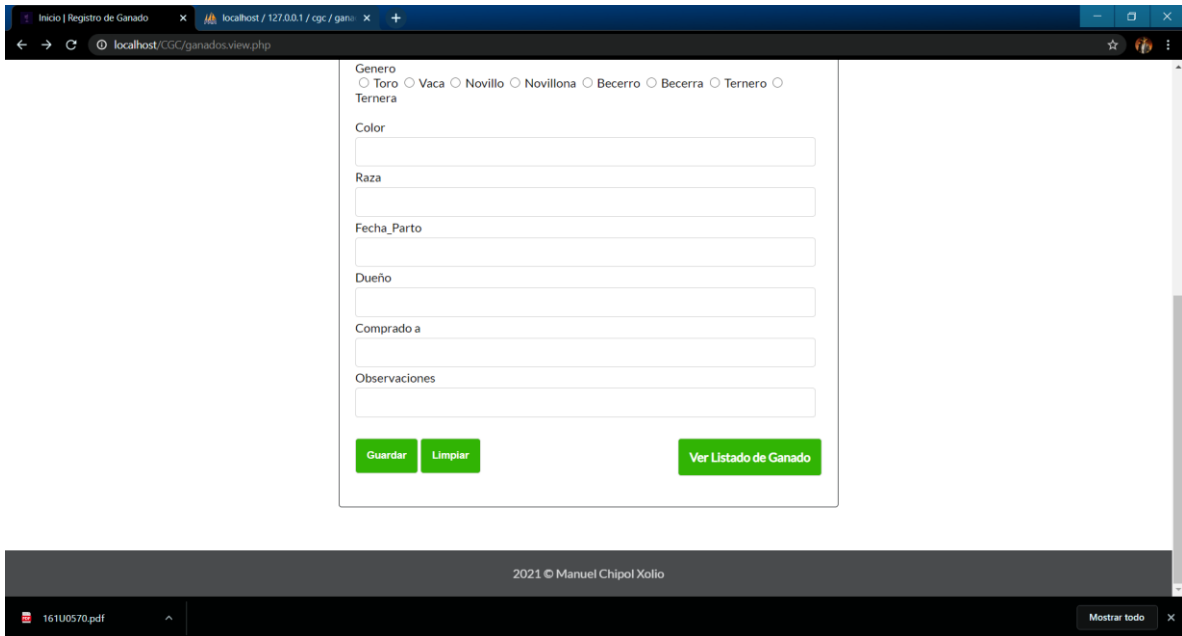


Fig. 4.2.4 Ventana registro de ganado

4.3 Conclusiones

La recolección de información de base de datos para la realización de un análisis exhaustivo, permitió plantear los diferentes requerimientos establecidos en el sistema de gestión de ganado para el Rancho “El Mirador”.

La base de datos lógicamente organizada y estructurada, facilitó la manipulación de datos permitiendo efectuar consultas, ingresos, usuarios etc.

En la codificación del software se realizaron pruebas para optimizar las funcionalidades requeridas al ingresar los datos, localizando las fallas y corregirlas acertadamente.

Una vez que el software fue ejecutado correctamente ya se podrían registrar los usuarios, de manera que ya podrían entrar al sistema, ya sea el administrador o el ganadero.

Recomendaciones

Una vez recolectado los datos, se deben de ordenar, clasificar y seleccionar los necesarios, para no tener exceso de información al momento de realizar los requerimientos.

En el diseño de la base de datos se deben emplear tablas y campos relevantes, aplicando políticas de seguridad para conservar la integridad de la información.

Para el desarrollo del sistema informático que en este caso es el control ganadero, debe de presentar una interfaz detallada de todos los procedimientos a realizar, para que personas inexpertas puedan ser uso de sus funcionalidades.

Verificar un correcto funcionamiento del software de todas las especialidades antes expuestas por el cliente, realizando pruebas contantes que ostenten su trabajo antes de la entrega total al final del proyecto.

Emplear una PC o una laptop con los requerimientos necesarios y apropiados que permita el óptimo funcionamiento del sistema con una mayor rapidez.

Experiencia

Este proyecto ha sido de gran ayuda para tomar experiencia en el ramo Informático, ya que trabajar en un entorno donde estas en comunicación con el cliente, es muy enriquecedor.

A manera muy personal puedo decir que este proyecto me ha dado un gran aprendizaje en los siguientes aspectos:

Comunicación: Es la base del proyecto, ya que, sin comunicación adecuada, ninguna de las mejoras hubiera podido ser realizada. Este proyecto nos ha ayudado mucho a buscar la mejor manera de expresar mis ideas ante los administrativos, como ante los colaboradores.

Trato hacia las personas: Un aspecto totalmente importante es que aprendimos a trabajar con personas y aprender de ellas para tener un mejor desempeño en mis actividades, al mismo tiempo que poder apoyar en las actividades de los demás.

Trabajo en equipo: El tener relación con el administrador y el encargado del rancho ha dado la oportunidad de reconocer la importancia del trabajo en equipo. Así mismo, gracias a ello, entendimos que lo más importante es involucrar a todos y hacerlos sentir parte de un proyecto tan importante. Esto genera confianza y la oportunidad de ser escuchado.

Toma de decisiones: Un aspecto de gran importancia para nosotros como ingenieros es la toma de decisiones, ya que estas influyen directamente al proyecto por lo que es necesario conocer el entorno de una decisión, así como sus posibles consecuencias para tomar la mejor posible.

Capacidad de análisis: Al estar dentro de procesos complejos, fue necesario desarrollar la habilidad de analizar la situación desde muchas perspectivas para una buena toma de decisión.

Manejo y selección de información: Al estar dentro de un proceso y de un análisis de datos, es necesario conocer la forma de procesar más rápido los datos para seleccionar aquellos que son necesarios.

Compromiso: Un elemento muy importante sin el cual, ninguna mejora es posible de realizar.

Responsabilidad: Estar a cargo de actividades nos hizo desarrollar un fuerte sentido de responsabilidad para con la empresa.

Competencias desarrolladas

Para el desarrollo del proyecto de residencia se consideraron las siguientes asignaturas: Metodología de Investigación, Taller de Investigación I y II, Fundamentos de Bases de Datos, Taller de Bases de Datos, Inteligencia de Negocios, Fundamentos de Sistemas de Información, Estrategias de Gestión Servicios de TI y Administración de Bases de Datos, que se aplicaron y sus competencias respectivas, de acuerdo a sus planes de estudio.

COMPETENCIA DESARROLLADA Y APLICADA	¿DÓNDE SE APLICÓ?
Capacidad de análisis y síntesis	Cuando se consultó bibliografía para justificar el proyecto.

Capacidad de organizar y planificar	Para establecer objetivos, plazos, definir prioridades y controlar la calidad del trabajo.
Capacidad de aplicar los conocimientos.	A través del desarrollo del proyecto
Capacidad de generar nuevas ideas.	Al momento de plantear el proyecto basándonos en las necesidades de la empresa, además de que no existe actualmente ningún trabajo similar al planteado
Habilidades para buscar, procesar y analizar información procedente de diversas fuentes	Para localizar información de diversas fuentes, indagar, buscar, analizar y procesar información

Referencia Bibliografía

Ávila Parra, K. (2015, 3 noviembre). Ganadería y medio ambiente. Ganadería y medio ambiente. <https://ganaderiaymedioambiente.wordpress.com/marco-teorico/>

Estela Raffino, M. (2020, 20 junio). base de datos. Concepto.de. <https://concepto.de/base-de-datos/>

Elías, J. (2008, 21 abril). Importancia de la ganadería en el desarrollo. desarrollo y defensa. <https://desarrolloydefensa.blogspot.com/2008/04/importancia-de-la-ganadera-en-el.html>

Red de Portales News Detail Page. (2020, 20 marzo). Los lenguajes de programación más usados en la actualidad. <https://www.universia.net/ar/actualidad/empleo/lenguajes-programacion-mas-usados-actualidad-1136443.html>

Villán, V. R. (2020, 27 noviembre). Qué son las metodologías ágiles y cuáles son sus ventajas empresariales. Thinking for Innovation. <https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/>

Luca, D. (2019, 10 octubre). Sublime Text: un editor de código multiplataforma. Damián De Luca. <https://damiandeluca.com.ar/sublime-text-un-editor-multiplataforma>

Agusti, R., Bernardo, F., Casadevall, F., Ferrús, R., Pérez, J. y Sallent, O. (2010). LTE: Nuevas Tendencias en Comunicaciones Móviles. Madrid: Fundación Vodafone España.

Ahonen, T., Barret, J. & Golding, P. (2002). Services for UMTS, Creating Killer Applications in 3G. West Sussex: John Wiley & Sons.

Holma, H. & Toskala, A. (2004). WCDMA for UMTS. Radio Access For Third Generation Mobile Communications. Inglaterra: John Wiley & Sons.

U.S. Department of health and human services. (2004). Seventh Report of the Joint National Committee on Prevention, Detection, Evaluation, and Treatment of High Blood Pressure. Recuperado de <http://www.nhlbi.nih.gov/guidelines/hypertension/jnc7full.pdf>.

Anexos

Carta de autorización



Península de moreno, Mpio. de Catemaco, Ver. 8 de enero 2020

Asunto: Oficio de Autorización de
Residencias profesionales

Lic. German Ventura Tenorio
Jefe del Departamento de Estudios Profesionales

Por medio del presente, autorizo que el C. MANUEL CHIPOL XOLIO, con numero de control 161U0570 de la carrera de INGENIERÍA INFORMÁTICA, desarrollara una "APLICACIÓN WEB PARA LA GESTIÓN DE GANADO BOVINO PARA EL RANCHO "EL MIRADOR"" de la comunidad antes mencionada, con fecha límite hasta el día 8 de enero del 2021.



MANUAL DE INSTALACIÓN DE APLICACIÓN

SOFWARE

Manuel Chipol Xolio | Manual | 15/12/2020

INDICE

INTRODUCCIÓN.....	2
INSTALACIÓN DE XAMPP.....	3
INSTALACION APLICACIÓN.....	3
EJECUTAR XAMPP.....	4
APLICACIÓN FUNCIONAL.....	6
CONCLUSIÓN.....	6

INTRODUCCIÓN

El presente documento tiene como objetivo instruir al usuario en proceso de instalación de aplicaciones open Source XAMPP ya que es una distribución de Apache completamente gratuita y fácil de instalar que contiene MariaDB, PHP y Perl. El paquete de instalación de XAMPP ha sido diseñado para ser increíblemente fácil de instalar y usar. Todo esto se requiere para desarrollar en entorno web.

El sistema funcionara bajo una red local, específicamente mediante una intranet. Al cual se accedera mediante un browser con la siguiente direccion:

<http://localhost/CGC/index.php>

los usuarios podran visualizar la informacion que estos requieran, sin embargo, no se les permitira modificar informacion que sea clasificada y administrada por un usuario de rol administrador.

User Administrador:

Usuario: admin

Contraseña: admin123

1.- Instalación XAMPP

1.1 Descargar el instalador de XAMPP para sistema operativo windows. Fig. 1.1



1.2 Copiar instalador en el disco local C y ejecutar como administrador para proceder con la instalacion.

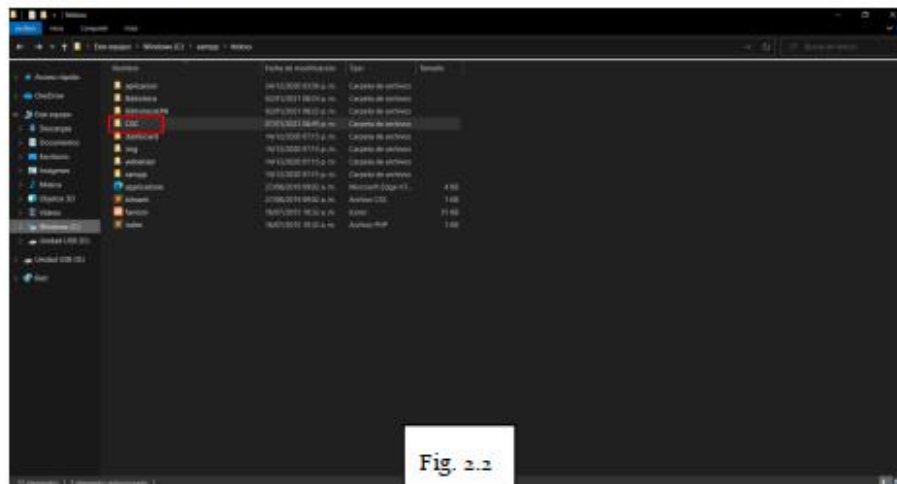
1.3 Debe de seguir las instalaciones hasta completar la instalacion.

2.- Instalacion Aplicación.

2.1 Identificar carpetas de archivos y base de datos.

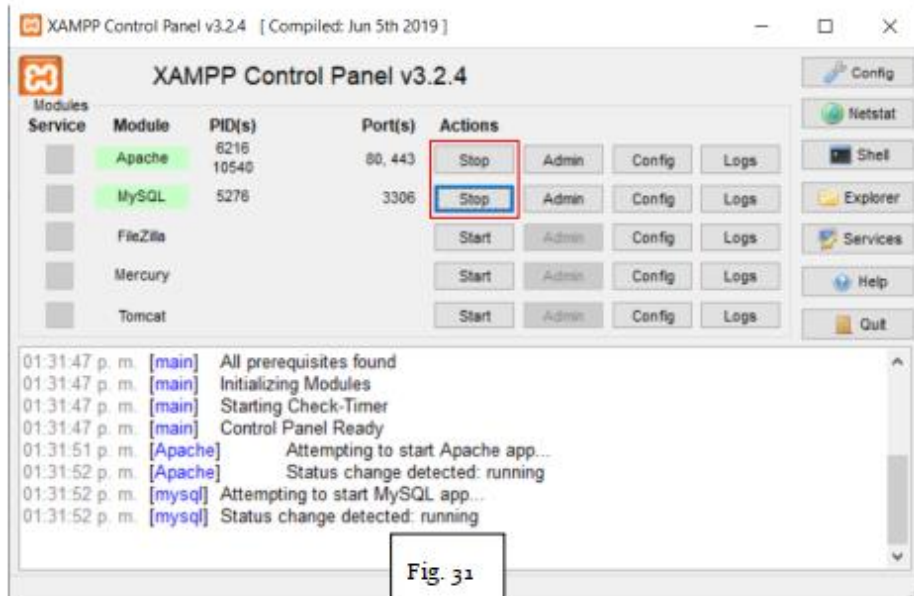
2.2 Dentro de la carpeta XAMPP creada en el disco C, debe pegar carpeta de archivos CGC tal y como se muestra en la direccion:

C:\xampp\htdocs\CGC

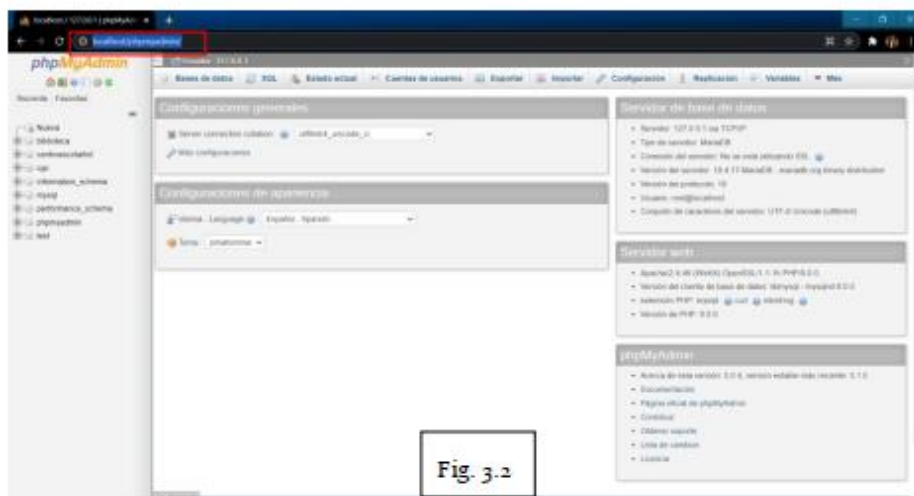


3.- Ejecutar XAMPP

3.1 Seleccionamos en actions APACHE Y MYSQL, tal y como se muestra en la figura.



3.2 Abrimos cualquier navegador ya se Chrome o explore y ejecutamos la siguiente dirección para cargar la BD <http://localhost/phpmyadmin/>, tal y como se muestra en la figura 3.2



3.3 Damos click en crear nueva base de datos, y damos asignar el nombre CGC y luego click en “Crear”. Tal y como se muestra en la figura 3.3.

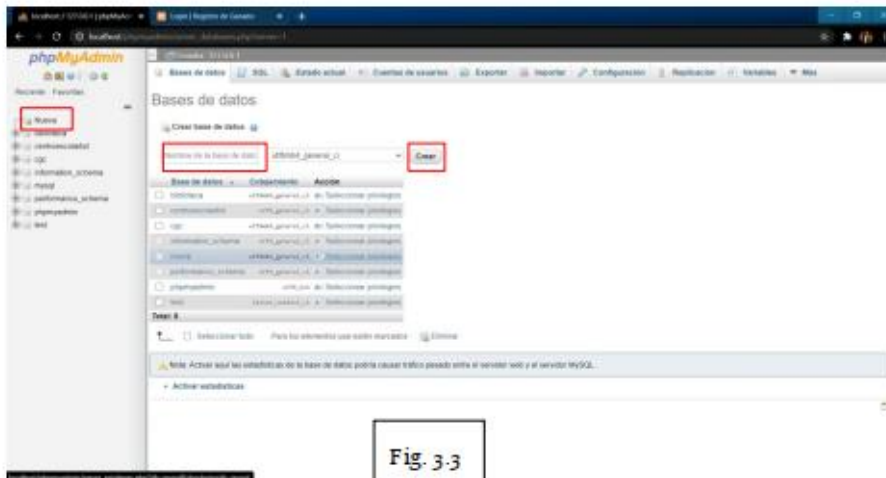


Fig. 3.3

3.4 Proceder a importar la base de datos, de manera que ya está seleccionado. Tal y como se muestra en la figura 3.4.

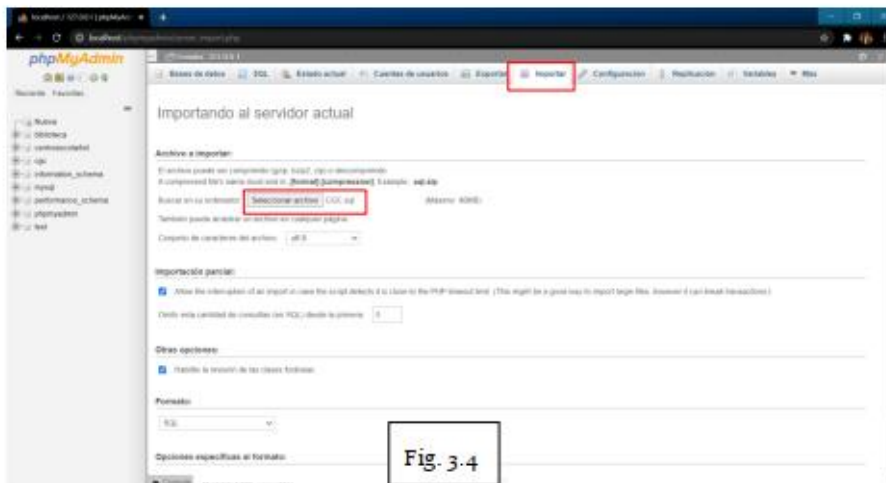


Fig. 3.4

4.- Aplicación Funcional

4.1 Ingrese al navegador ya antes mencionado en la siguiente dirección <http://localhost/CGC/index.php> para acceder a la aplicación. De manera que pedirá el usuario y la contraseña. Fig. 4.1



CONCLUSION

En conclusión, para que este sistema funcione correctamente debe de tener instalado los siguientes programas así como xampp para sistemas operativos Windows y Ubuntu y así poder proceder con cargar el sistema en esta dirección `C:\xampp\htdocs`, de manera que así podrá correr el sistema correctamente. Tal y como se muestra en este manual de instalación.