



TECNOLÓGICO
NACIONAL DE MÉXICO®

INSTITUTO TECNOLÓGICO DE PABELLÓN DE ARTEAGA.



Herramienta de Transformación Bidireccional de
Modelos de Entidad-Relación a Diseño de Base de
Datos Relacionales

REPORTE FINAL PARA ACREDITAR LA RESIDENCIA PROFESIONAL DE
LA CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN
Y COMUNICACIONES

PRESENTA

Edith Martínez Martínez

TUTORES

Benito Rodríguez Cabrera.
Nivia Iracemi Escalante García.

PABELLÓN DE ARTEAGA, AGS., DICIEMBRE 2022



TECNOLÓGICO
NACIONAL DE MÉXICO®



INSTITUTO TECNOLÓGICO®
de Pabellón de Arteaga

ITEC

INSTITUTO TECNOLÓGICO DE PABELLÓN DE ARTEAGA.

SUBDIRECCIÓN ACADÉMICA

INGENIERÍA TECNOLOGÍAS DE LA INFORMACION Y

COMUNICACION

TESIS

REPORTE FINAL PARA ACREDITAR
LA RESIDENCIA PROFESIONAL DE
LA CARRERA DE INGENIERÍA EN
TECNOLOGÍAS DE LA INFORMACIÓN
Y COMUNICACIONES

PRESENTA

EDITH MARTÍNEZ MARTÍNEZ

TUTORES

BENITO RODRÍGUEZ CABRERA.

NIVIA IRACEMI ESCALANTE GARCÍA.

PABELLÓN DE ARTEAGA, AGS., DICIEMBRE 2022

AGRADECIMIENTOS

Expreso mis más sinceros agradecimientos principalmente al LIA, por darme la oportunidad de colaborar con ellos en todo momento. A mis asesores la Doctora Nivia Escalante por guiarme y motivarme en todo momento del proceso de residencias, al maestro Benito Rodriguez por impulsarme, apoyarme y darme las herramientas y consejos para motivarme a seguir adelante.

Al TecNM Campus Pabellón de Arteaga por abrirme las puertas de esta institucion, para seguir con mi formación profesional.

Desde mi corazón agradezco a mis padres, gracias por tener la suficiente paciencia en estos años en los cuales vieron diferentes facetas de su hija, para culminar su carrera profesional.

Agradezco mis amigos más cernanos, que estuvieron en estos años acompañandome en el proceso y por ayudarme y regalarme los mejores recuerdos de la carrera.

RESUMEN

Herramienta de Transformación Bidireccional de Modelos de Entidad-Relación a Diseño de Base de Datos Relacionales

Por: Edith Martínez Martínez

En el presente proyecto está enfocado en atacar problemas de realizar diagramas de forma manual, obteniendo versiones creadas que pasan por alto los análisis de datos, requerimientos y reglas de los modelos de datos, provocando un mal desarrollo de trabajos a futuro, proyectos o sistemas que se basan de estos diagramas. Realizando una investigación y un análisis en las reglas de los modelos de datos, su impacto en el proceso de desarrollo y de almacenamiento en un sistema informático. Es por eso que se realizó el desarrollo de un sistema de Conversión de diagramas de modelos de datos, así como su nombre lo indica su funcionamiento será convertir o transformar los datos que el usuario requiera y forme en un diagrama E-R, al diagrama relacional, desarrollando la interfaz principal del sistema con entornos JavaScript, React.js, librerías para Visual Studio Code, librerías para lienzo y figuras. Obteniendo un sistema e interfaz con un diseño que de primera mano será amigable a simple vista, fomentando la interacción Humano-Computadora y aplicando el ser desarrollado en entornos diferentes a los ya conocidos. Desarrollando competencias de desarrollo analítico, desarrollo de procedimientos, fomentación de análisis de datos y requerimientos.

ÍNDICE GENERAL

Agradecimientos	III
Resumen	IV
1. Generalidades del proyecto	1
1.1. Antecedentes	1
1.1.1. Desarrollo de herramienta CASE para el diseño de D. E-R Extendido.	1
1.1.2. Plugin para Generar Esquemas Dimensionales a partir del Diagrama Entidad-Relación desde el Visual Paradigm.	2
1.1.3. Sitio Web Aprender Modelo Entidad-Relación	2
1.2. Introducción	3
1.3. Descripción de la empresa u organización y del puesto o área del trabajo del residente.	3
1.4. Problemas a resolver priorizandolos	5
1.5. Justificación	6
1.6. Objetivos	6

1.6.1. Objetivo general	6
1.6.2. Objetivos Especificos	6
2. Marco teórico	8
2.1. Marco Teórico	8
2.1.1. Sistemas de Información	8
2.1.2. Base de datos	9
2.1.3. Modelo de Datos	9
2.1.4. Modelo Entidad-Relacion	10
2.1.5. Modelo Relacional	12
2.1.6. Node.js	12
2.1.7. Visual Studio Code	12
2.1.8. Canvas	13
2.1.9. React.js	13
2.1.10. JavaScript	13
2.1.11. HTML	14
2.1.12. CSS	14
2.2. Trabajos relacionados	14
2.2.1. Power Designer	14
2.2.2. ERDPlus	16
3. Desarrollo	17

3.1. Procedimiento y descripción de las actividades realizadas.	17
3.1.1. Metodología de Prototipado	18
3.2. Recolección y refinamiento de requerimientos	19
3.2.1. Recolección de requerimientos	19
3.2.2. Diseño Rápido	20
3.2.3. Colores para Aplicación	25
3.2.4. Mockups	26
3.3. Optimización del Diseño	27
3.3.1. Construcción del Prototipo	27
3.4. Evaluación del Diseño y Prototipo	30
3.4.1. Evaluación del Prototipo por el Cliente	30
3.4.2. Refinamiento del Prototipo.	31
3.5. Cronograma	32
4. Resultados	33
4.1. Resultados obtenidos	33
4.1.1. Producto de Ingeniería	33
5. Conclusiones	36
5.1. Conclusiones del proyecto	36
6. Competencias desarrolladas	37

6.1. Competencias desarrolladas y/o aplicadas	37
A. Anexos	38
A.1. Anexos	38
A.2. Registros de productos o marcas	38

ÍNDICE DE FIGURAS

1.1. Organigrama. Fuente: Propia	5
2.1. Modelo de datos. Fuente: [1]	10
2.2. Entidad-Relacion. Fuente: [2]	10
2.3. Componentes E-R. Fuente:[3]	11
2.4. Pantalla Power Designer. Fuente: [4]	15
2.5. Pantalla de Conversión ERDPlus. Fuente: [5]	16
3.1. Metodología de Prototipado. Fuente: Propia	18
3.2. Diagrama de casos de uso. Fuente: Propia	19
3.3. Crear y Convertir. Fuente: Propia	21
3.4. Descargar Conversion. Fuente: Propia	22
3.5. Administración. Fuente: Propia	23
3.6. Acciones del Sistema. Fuente: Propia	24
3.7. Actividades Básicas. Fuente: Propia	25
3.8. Colores para sistema. Fuente: Propia	26

3.9. Iniciar sesión. Fuente: Propia	26
3.10. Diseño de Interfaz. Fuente: Propia	27
3.11. Versiones node. Fuente: Propia	28
3.12. Vista Práctica. Fuente: Propia	28
3.13. Terminal Npm Iniciada. Fuente: Propia	29
3.14. Cronograma. Fuente: Propia	32
4.1. Interfaz Principal. Fuente: Propia	33
4.2. Botones y lienzo. Fuente: Propia	34
4.3. Diagrama terminado. Fuente: Propia	34

ÍNDICE DE TABLAS

3.1. Prueba 1.	30
3.2. Prueba 2.	31

CAPÍTULO 1

GENERALIDADES DEL PROYECTO

1.1 ANTECEDENTES

A continuación, se muestran antecedentes de herramientas que basaron su desarrollo enfocado en los modelos de base de datos, como también la forma en como los implementaron.

1.1.1 DESARROLLO DE HERRAMIENTA CASE PARA EL DISEÑO DE D. E-R EXTENDIDO.

El autor desarrolló una herramienta CASE para facilitar el proceso de mapeo en el Modelo E-R y E-R Extendido para estudiantes, se pretendió poner a disposición la herramienta para apoyar su proceso de aprendizaje. Fue desarrollada dentro del Front-end. Su disponibilidad está ubicada en línea para beneficio de los estudiantes del sector técnico superior como nivel licenciatura. La herramienta CASE que se utilizó (por sus siglas en inglés Computer Aided Software Engineering, Ingeniería de Software Asistido por Computadora) aumentando la productividad y la calidad en el desarrollo de software, así como la automatización de las actividades involucradas [6].

1.1.2 PLUGIN PARA GENERAR ESQUEMAS DIMENSIONALES A PARTIR DEL DIAGRAMA ENTIDAD-RELACIÓN DESDE EL VISUAL PARADIGM.

Visual Paradigm fue el equipo de desarrollo de software para realizar análisis y diseños de sistemas con facilidad. Este proporcionó productos que facilitaron las organizaciones y diseñar visualmente en forma esquemática, es decir diagramas, integrar y desplegar aplicaciones de la misión crítica de la empresa. Visual Paradigm se encontró entre las más usadas, esta puede crear ingeniería inversa del código de diagramas como también ofreció ida y vuelta para diversos lenguajes de programación. Además de que trabaja con multiplataforma como Java y función para Sistemas Operativos como Windows, Linux, Mac Os [7].

1.1.3 SITIO WEB APRENDER MODELO ENTIDAD-RELACIÓN

El autor explica que las TICs en el ámbito escolar es un gran reto, ya que al parecer los docentes deben integrar las tecnologías de una manera equilibrada para fomentar el aprendizaje y el desarrollo, el sitio web Aprender Modelo Entidad-Relación fomenta el aprendizaje para la base de datos, facilitar y mejorar habilidades de los estudiantes, la aplicación está basada en contenidos de los modelos o retículas de aprendizaje, además que está adaptado a objetivos de asignaturas específicas y cuenta con información actualizada para estudiantes. El sitio web fue elaborado con Joomla de desarrollo web para entrar en clasificación de Sistema de Administración de Contenidos [8].

1.2 INTRODUCCIÓN

La idea principal del proyecto es desarrollar un sistema conversor de diagramas E-R al modelo relacional, con la finalidad de implementar la optimización de crear diagramas racionales considerados para proyectos como Entidad-Relación y así facilitar modelar los datos, como también optimizar tiempos, en la realización de estructura y análisis de datos para un proyecto que implique el procesamiento de datos a través de diagramas relacionales para base de datos. Abarcando el ayudar con aprendizajes y familiarizar a principiantes con el desarrollo de sistemas de recolección de datos, así como ver la comparativa y como se convierten las relaciones de datos en el diagrama relacional. Se pretende fomentar el aprendizaje por su nivel básico de utilización, para que personas sin dominio de temas como diagramas relacionales y diagramas E-R, puedan crear, experimentar y analizar datos antes y después de la conversión.

Se creará una investigación de los conceptos de los que se enfocara el proyecto, sus métodos y las reglas a seguir, de la misma manera seleccionar las herramientas que se van a utilizar en la construcción del sistema. Se presentarán las versiones de los entornos que se instalaron y como se activó el local host dentro de una terminal y los resultados que se obtuvieron del trabajo conjunto de los entornos con los que se desarrolló.

1.3 DESCRIPCIÓN DE LA EMPRESA U ORGANIZACIÓN Y DEL PUESTO O ÁREA DEL TRABAJO DEL RESIDENTE.

El Laboratorio de Iluminación Artificial (LIA), con ubicación en el Instituto Tecnológico de Pabellón de Arteaga, Dirección: carretera a la estación de Rincón de Romos, Km. 1. C.P.20670, en el municipio de Pabellón de Arteaga, Aguascalientes.

El LIA fue fundado en el 2017 por el Dr. Ernesto Olvera González y la Dra. Nivia Iracemi Escalante García, dedicado al estudio y caracterización de la luz y su efecto en productos del sector agroalimentario, contando a la fecha con cuatro líneas de investigación y desarrollo tecnológico:

- Diseño, Construcción e Implementación de Sistemas de Producción Multinivel en espacios cerrados con luz artificial LED que permiten potenciar el contenido nutrimental de cultivos como microgreens, lechuga, lenteja, espinaca, albahaca, alfalfa, entre otras; a través de recetas de luz LED con radiación ultravioleta (diferentes combinaciones de color).
- Desarrollo de sistemas de desinfección de alimentos con radiación ultravioleta LED tipo A, B y C aplicados a productos agroindustriales y en fresco.
- Implementar estrategias tecnológicas con luz artificial tipo LED para preservar, extender y/o acelerar el tiempo de vida en almacén de frutas y verduras, además de analizar el efecto sobre la biosíntesis de compuestos (licopeno, capsaicina, entre otros) durante su estancia en anaquel.
- Investigación, integración y aplicación de la agricultura de precisión con el uso de tecnología aérea no tripulada (VANTs) para el monitoreo y detección de plagas en diferentes cultivos con el objetivo de evitar pérdidas en la producción.

Investigación, Integración y Aplicación de la agricultura de precisión con el uso de tecnología aérea no tripulada (VANTs) para el monitoreo y detección de plagas en diferentes cultivos con el objetivo de evitar pérdidas en la producción.

El laboratorio está conformado de la siguiente manera, Figura 1.1

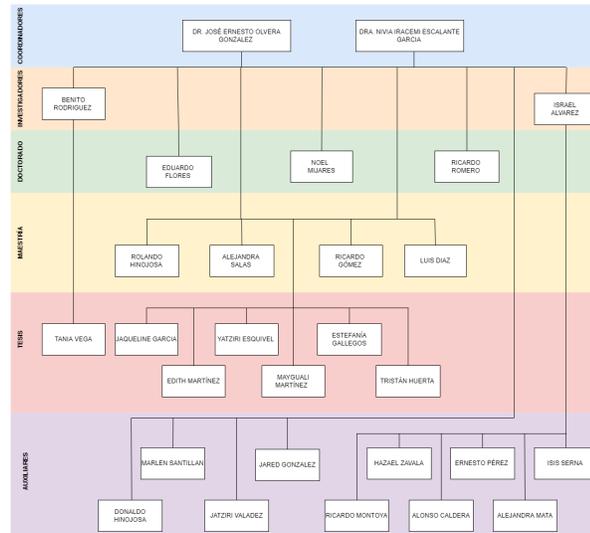


Figura 1.1: Organigrama. Fuente: Propia

1.4 PROBLEMAS A RESOLVER PRIORIZANDOLOS

La creación de diagramas relacionales se implementan o se consideran para trabajos, proyectos o sistemas, los alumnos o programadores optan por realizar esos diagramas de manera manual o dentro de un gestor de BD (Base de Datos), mismos que se crean recurriendo a ejemplos de diseño, relaciones o similitudes, crear versiones variadas de los mismos diagramas, generando errores relacionales, al pasar por alto en su mayoría el análisis de requerimientos, las reglas de los modelos Entidad-Relación y de los Modelos Relacionales. Este tipo de situaciones provoca pérdidas de tiempo, desorientación, resultados diferentes, es decir, al buscar un relacionamiento funcional se pierde visión del objetivo inicial.

Otro de los problemas que se tienen son la falta de herramientas que simplifiquen el proceso de conversión de los modelos Entidad-Relación a Relacional, respetando las reglas y la teoría de Base de Datos.

1.5 JUSTIFICACIÓN

El crear un sistema de conversión de modelos Entidad-Relación a Relacional permitirá ofrecer a los usuarios el tener la opción de desarrollar los tipos de modelos de datos. Sirviendo de apoyo para los usuarios en el modelado de datos, es decir, facilitar la creación de sus diagramas relacionales, desarrollando como tal la optimización de tiempos en la creación de dichos diagramas sin excluir el uso de las herramientas y técnicas de base de datos. Resolviendo problemas como el modelado de datos redundante, dando la oportunidad de que expertos y principiantes se hagan de esta herramienta como forma de verificar o agilizar los procesos. Ofreciendo método de enseñanza al público al que va dirigido.

1.6 OBJETIVOS

1.6.1 OBJETIVO GENERAL

Desarrollar un sistema de conversión de modelos de datos tipo Entidad-Relación a esquemas relacionales para optimización de resultados para Bases de Datos.

1.6.2 OBJETIVOS ESPECIFICOS

- Analizar las reglas de conversión de modelos Entidad- Relación a relacionales.
- Analizar los requerimientos funcionales de la plataforma en cuanto a los procesos de conversión.
- Crear arquitectura funcional de los componentes de diseño.

-
- Desarrollar la funcionalidad del sistema utilizando JavaScript, HTML, React.js y Canvas.
 - Evaluar los productos obtenidos a través de varias pruebas.

CAPÍTULO 2

MARCO TEÓRICO

En el siguiente capítulo se mencionará algunos de los conceptos y herramientas que se ocuparan a lo largo de este proyecto. Mencionando también aplicaciones relacionadas o que son similares a este proyecto.

2.1 MARCO TEÓRICO

2.1.1 SISTEMAS DE INFORMACIÓN

En la actualidad es de suma importancia contar con Sistemas de Información, con el fin de satisfacer las necesidades de las organizaciones, ya sea el plano laboral y estudiantil.

En los sistemas de información se puede mostrar la interacción entre algunos elementos como lo son equipo computacional, recurso humano, flujo de datos, información, telecomunicaciones, entre otros. Las características que identifican a los Sistemas de Información, está el procesamiento de entradas de información, almacenamiento, procesamiento para la transformación de datos y salidas de información [9].

Para tener un sistema de información eficiente y asegurar tener un correcto respaldo y organización de la información recolectada y almacenada, se debe contar con herramientas que se encargan de esto, como lo son:

2.1.2 BASE DE DATOS

Al ser un gestor de base de datos puede encargarse de almacenar las actividades y datos recabados dentro de un sistema de información, organizándola y relacionándola como marque su modelo relacional ya establecido. Dicha información recaudada podrá ser maniobrada por los administradores que controlen la Base de Datos. Al involucrarse las bases de datos en este tipo de proyecto, se requerirá sus reglas y delimitaciones para la conversión al modelo relacional del gestor [10].

Dentro de las bases de datos existe el modelado que conlleva reglas y funciones para hacer que los datos puedan relacionarse, hacer conjuntos y obtención de varios datos en una sola consulta, a este modelado se le dá el nombre de modelos de datos.

2.1.3 MODELO DE DATOS

Los modelos de datos se encuentran bajo la estructura de la Base de Datos, siendo un equipo de herramientas para la descripción de datos, relación entre los mismos, similitud y restricciones, siendo la representación conceptual de un modelo de datos visualizado en la Figura 2.1. Existen varios tipos de modelos de datos, como lo es el modelo Orientado a Objetos, Dimensional, Jerárquico y por último el modelo E-R y el modelo Relacional [11].

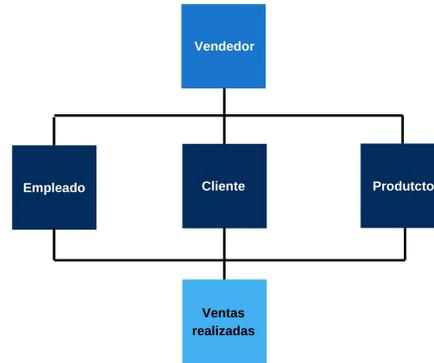


Figura 2.1: Modelo de datos. Fuente: [1]

2.1.4 MODELO ENTIDAD-RELACION

Los modelos de datos Entidad-Relación son parte del sistema que se construirá y representará los datos mediante un esquema conceptual, planteando el significado de los mismos y su interacción representándose como la Figura 2.2. Su popular uso y utilidad hacen que sea un método para el diseño de bases de datos, ayudando a los usuarios a que sea de mayor entendimiento, basándose en sus conceptos [3].

Modelo entidad-relación

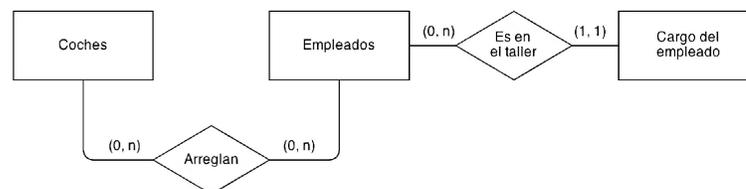


Figura 2.2: Entidad-Relacion. Fuente: [2]

Características que conforman el modelo E-R.

- Entidad: cosa u objeto que se diferencia de los demás, resaltando como el componente representativo de los demás datos.

- Atributo: conjunto que representa las características o propiedades de una entidad, los atributos también pueden tener atributos dentro de los mismos, llevándolo a ser atributos compuestos.
- Atributos compuestos: los atributos compuestos se dividen en sub-partes, es decir, como un ejemplo un atributo es el *nombre-cliente* siendo dividido en varios componentes que es la estructuración separada del nombre (Nombre, primer apellido, segundo apellido). Al igual los atributos pueden llegar a tener un valor nulo, esto sucede cuando la entidad del atributo no le tiene valor, es decir, no llega a ser aplicable en casos especiales.

Además de las entidades y atributos que son los principales que conforman un modelo entidad-relación, para crear un diagrama de datos requiere los siguientes elementos
 Figura 2.3.

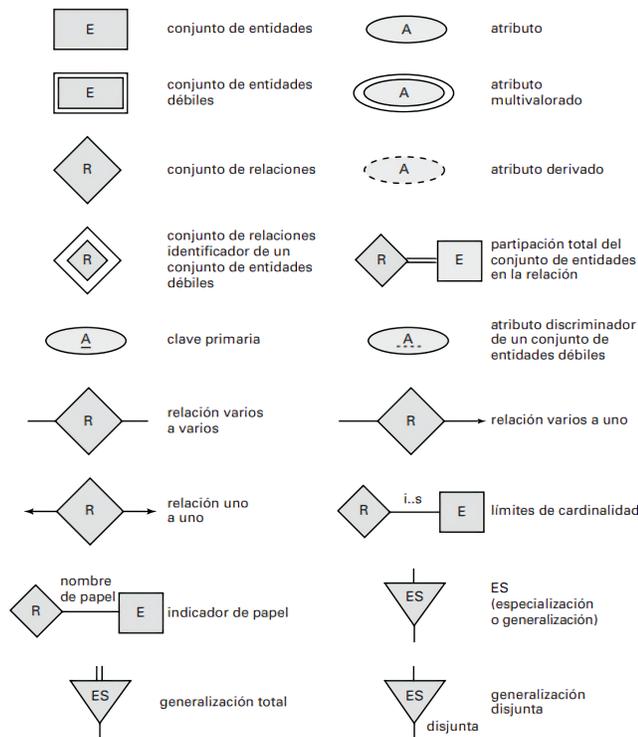


Figura 2.3: Componentes E-R. Fuente:[3]

2.1.5 MODELO RELACIONAL

El modelo relacional es uno de los principales modelos para el procesamiento de datos, siendo similar en simplicidad con el modelo E-R, involucrando lenguaje de consulta, de manera representativa los modelos de entidad de relación a diferencia del ya mencionado, representado por tablas, conjunto de tablas compuestas por columnas y cada una su respectivo nombre y las relaciones entre sí [11].

En la parte del funcionamiento, diseño y lo visual se utiliza herramientas de código abierto que especificará y desarrollará el back-end del sistema. Para el desarrollo en el back-end se utilizaron aplicaciones y librerías específicas como lo son las siguientes.

2.1.6 NODE.JS

Entorno de tiempo de ejecución de JavaScript, por eso incluye el termino "Js" por el lenguaje mencionado, tiene la ventaja de incluir lo necesario para ejecución en tiempo real de un programa escrito en JavaScript. Utilizado como un modelo de entrada y salida de respuestas sin bloqueo, siendo de gran utilidad para la creación de aplicaciones de red rápidas [12].

Node al ser instalado, será utilizado al ejecutarse una terminal dentro de Visual Studio Code.

2.1.7 VISUAL STUDIO CODE

Aplicación de Microsoft, tipo software libre y de multiplataforma, editor de código fuente para desarrollo, es una de las plataformas compatible con diferentes S.O., con extensiones, herramientas y disponibilidad de ejecutar cualquier tipo de lenguaje

como lo son: JavaScript, TypeScript, por mencionar algunos. La herramienta de Visual Studio tendrá instalado ciertas librerías para facilitar el desarrollo del código del cual son las librerías de Canvas y React.js para JavaScript [13].

Visual Studio Code, es la sede de herramientas que se ocuparan en el desarrollo, entre ellas las librerías como la siguiente.

2.1.8 CANVAS

La librería Canvas, es utilizada para el desarrollo de líneas, figuras, dibujar gráficos igual para la adaptación de animaciones, imágenes y videos, su composición tanto simples como complejas [14].

2.1.9 REACT.JS

Librería de Open Source en JavaScript para interfaces de usuario, utilizado para React en el desarrollo de aplicaciones, como Facebook, Instagram y el servicio web de WhatsApp entre más aplicaciones populares. React es utilizado por los componentes, por la opción de que al crear los componentes son independientes y reutilizables [15].

2.1.10 JAVASCRIPT

Lenguaje de programación de mecanismos para seguimiento de órdenes, realización de tareas y repeticiones. El lenguaje de JavaScript trabaja con etiquetas de HTML y CSS.

Como la mayoría de los lenguajes de programación, éste asigna un tipo de dato al declarar variables, los tipos de datos que abarca JavaScript tenemos: number (valor numérico), BigInt (valor numérico grande), String (Valor de texto, caracteres),

Boolean(valor booleano), Function (funcion guardada en una variable), Object (objeto de estructura compleja) [16].

2.1.11 HTML

Lenguaje de código abierto mayormente utilizado para el desarrollo de las páginas web, sistemas y programas permitiendo mostrar al usuario a través de código, textos, imágenes, sonidos e inclusive videos, mostrados en cualquier navegador web o sin la necesidad de utilizar internet [17].

2.1.12 CSS

El lenguaje CSS nos permite darle a los anteriormente mencionados (texto, imágenes, sonidos y videos) de HTML el tamaño y diseño que requiera el desarrollador o el cliente, dando la definición de la presentación y apariencia visual [17].

2.2 TRABAJOS RELACIONADOS

Las herramientas relacionadas que pueden llegar a convertir los diagramas después del modelado son las siguientes.

2.2.1 POWER DESIGNER

Power Designer es una herramienta de diseño rápido dirigido a desarrolladores de sistemas de información, con el fin de diseñar estructuras para base de datos relacionales, con especificaciones, definición de información, tipo de dato y su

relación. Tiene como características el utilizar el modelado de procesos usando la tecnología de modelado de datos en los diagramas de flujo, diagramas UML estándar (caso, uso, clase, actividad, componentes). La definición de datos para Microsoft SQL Server, MySQL, modelado mediante especificaciones del usuario [18].

Mostrando en aplicación la creación de los modelos conceptual-lógico a físico dentro del modelado rápido con herramientas básicas que se requieren para su creación mostrado en la Figura 2.4, siendo la emulación de la conversión del Entidad-Relación al Relacional [4].

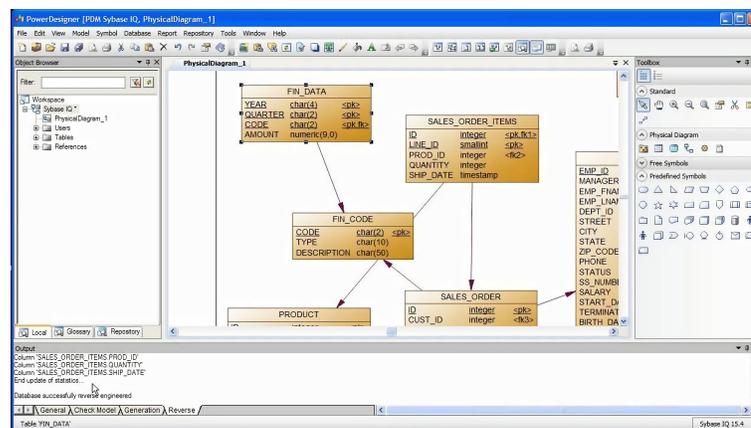


Figura 2.4: Pantalla Power Designer. Fuente: [4]

2.2.2 ERDPLUS

Herramienta básica para el modelado de datos para crear diagramas Entidad-Relación, diagramas relacionales, esquemas tipo estrella, declaraciones y relaciones; permitiendo tener la opción de conversión, entre otros. Como se muestra en la Figura 2.5 la conversión del diagrama Entidad-Relación al modelo relacional dentro de la aplicación ERDPlus [5].

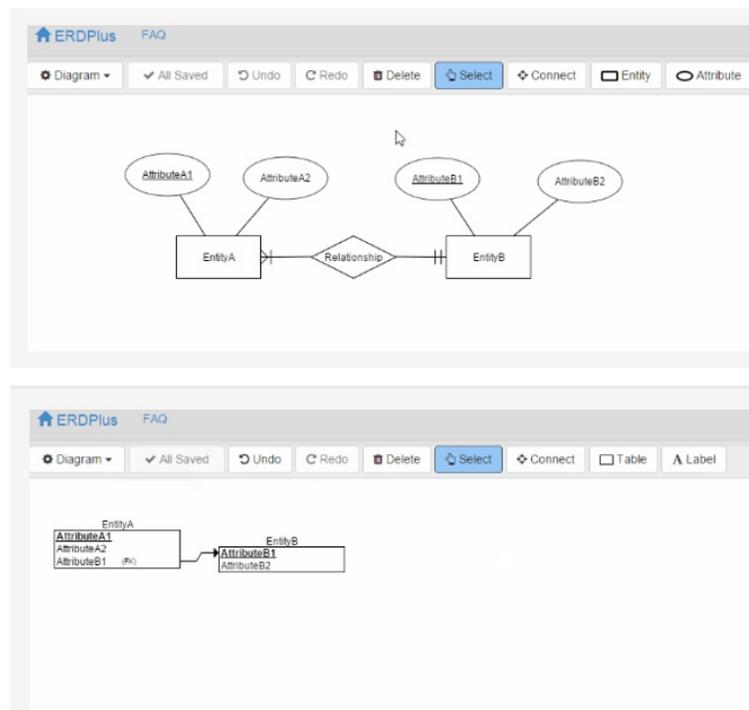


Figura 2.5: Pantalla de Conversión ERDPlus. Fuente: [5]

CAPÍTULO 3

DESARROLLO

3.1 PROCEDIMIENTO Y DESCRIPCIÓN DE LAS ACTIVIDADES REALIZADAS.

En la construcción de proyectos se requiere utilizar metodologías para la organización, asignación de actividades con la finalidad de tener un proceso organizado. La metodología de prototipado se considera como independiente, es decir, no pertenece a las metodologías ágiles y tradicionales.

En la construcción y organización del sistema se optó por la metodología ya mencionada, para control de las actividades y de los procesos, que se pueda adaptar a las necesidades del programador.

3.1.1 METODOLOGÍA DE PROTOTIPADO

La metodología de prototipado (por su nombre en inglés: prototyping) en ingeniería de software, representa de forma limitada del diseño de un producto. Involucrando fases para completar en el proceso de desarrollo. En la Figura 3.1 se muestran las fases a completar a lo largo de la construcción de proyecto, como son las fases de Recolección de requerimientos, Diseño rápido, Construcción del prototipo, Evaluación del Prototipo por parte del cliente, Refinamiento del Prototipo, Producto de Ingeniería.

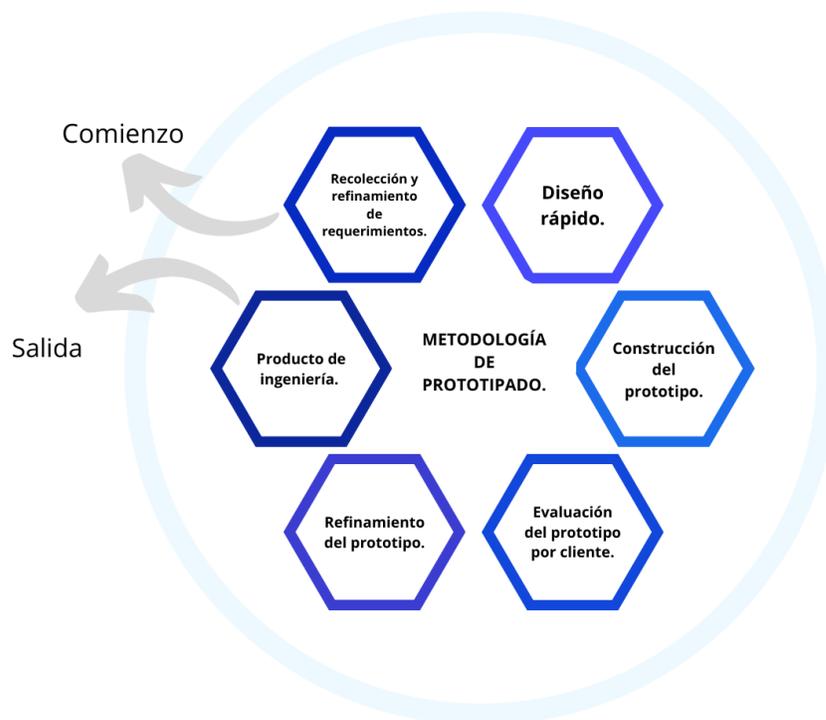


Figura 3.1: Metodología de Prototipado. Fuente: Propia

3.2 RECOLECCIÓN Y REFINAMIENTO DE REQUERIMIENTOS

3.2.1 RECOLECCIÓN DE REQUERIMIENTOS

Esta etapa se realizaron las reuniones de comunicación con el cliente, por las que se obtuvieron las características y requerimientos para la definición del funcionamiento del sistema.

Lo principal será la función en caso de que el autor (usuario general) de la Figura 3.2 quiera usar el sistema para conversión de diagramas.

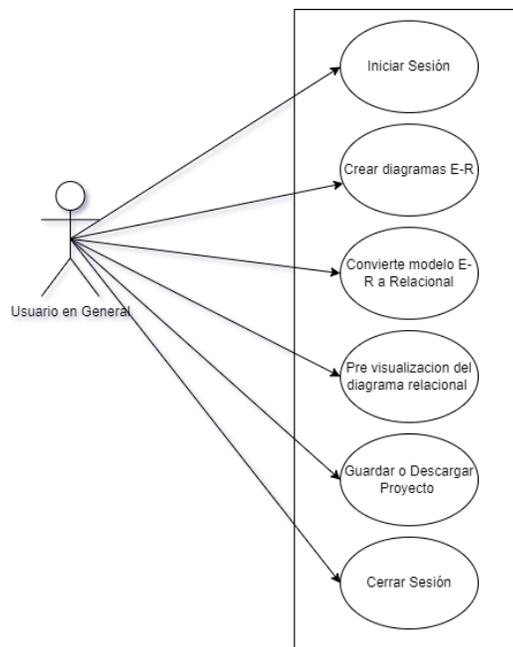


Figura 3.2: Diagrama de casos de uso. Fuente: Propia

3.2.2 DISEÑO RÁPIDO

Como primera etapa se tuvo en cuenta un curso para el desarrollo, de igual manera se realizaron investigaciones necesarias como herramientas para plantear más adelante.

- Se contempló especificar el funcionamiento del sistema a través de diagramas de actividades, diagramas de casos de uso, y diagramas de procesos.
- Se realizó la construcción del diseño rápido, es decir, crear una primera versión o un borrador del sistema, contemplando el plantear el funcionamiento.
- El diseño como primera versión del sistema se tomaron en cuenta las técnicas de interacción humano-computadora, con la finalidad de que el usuario tomando en cuenta los expertos y con principiantes tengan la facilidad de usarlo sin la necesidad de un manual de usuario.
- **Diagramas**

El sistema permitirá al usuario crear y convertir los diagramas que el desee desarrollar, llevando un proceso de creación y conversión, como se muestra en la Figura 3.3. Si el usuario desea crear un diagrama debe iniciar sesión con su perfil en sistema, al acceder al sistema podrá crear un diagrama E-R, de la cual tendrá la libertad de utilizar los componentes que requiera, asignar un nombre a cada entidad o agregar descripción al atributo. Al terminar el usuario tendrá en opciones la conversión de dicho diagrama.

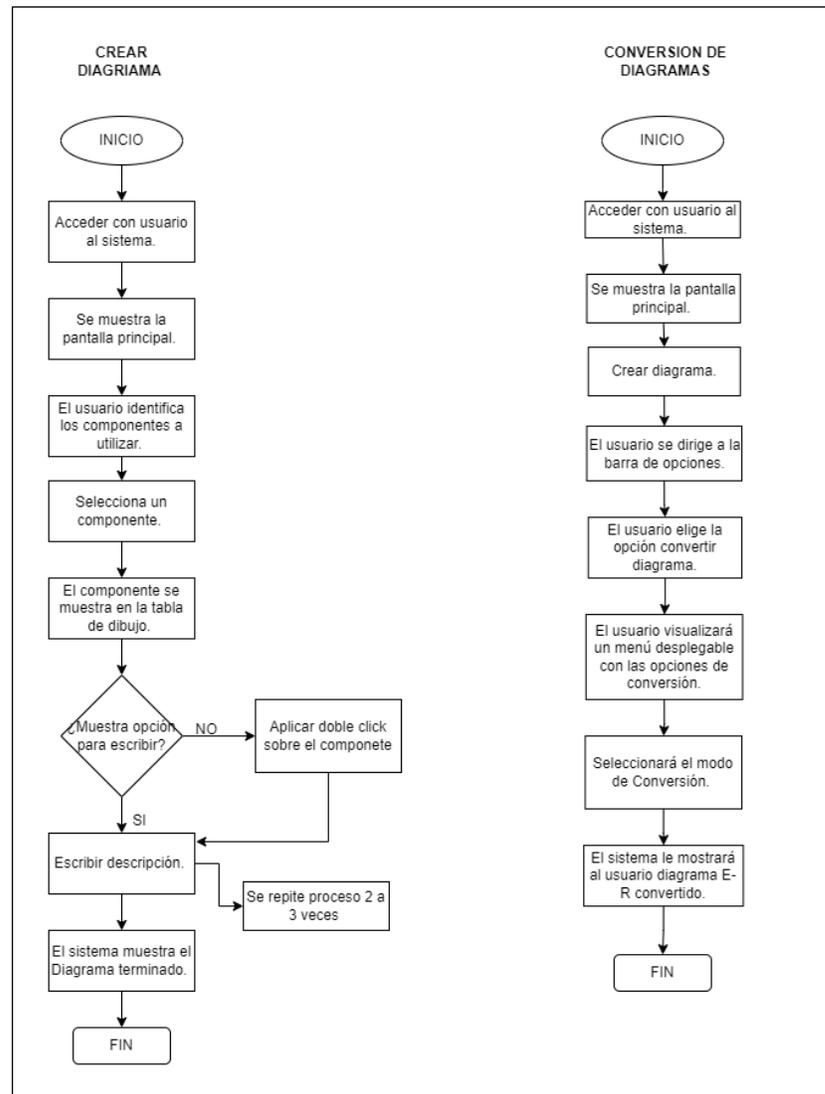


Figura 3.3: Crear y Convertir. Fuente: Propia

Al hacer la conversión del diagrama el usuario puede descargarlo, dirigiéndose a la barra de opciones y seguir las instrucciones como se marca en la Figura 3.4, guardándolo en una carpeta asignada.

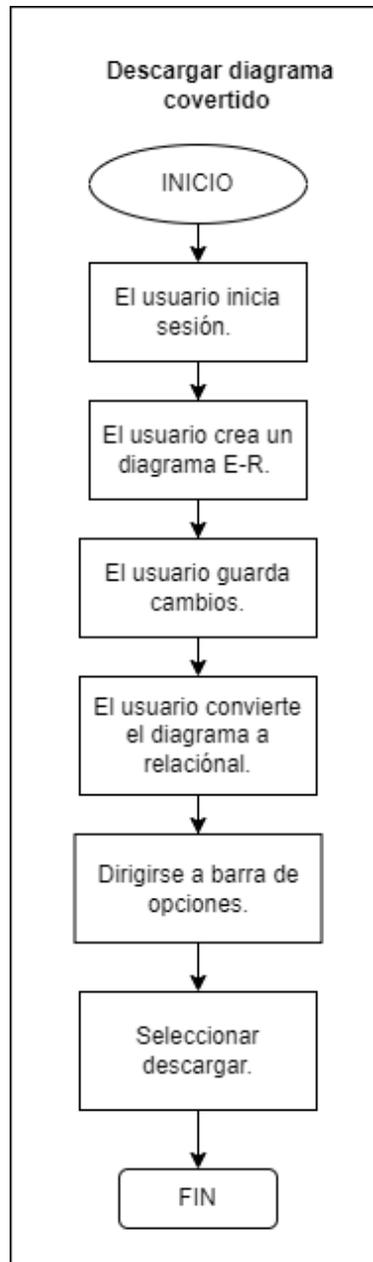


Figura 3.4: Descargar Conversion. Fuente: Propia

Para la administración de los proyectos, si el usuario ejerció la acción de guardar un diagrama para recurrir a él más tarde; si el usuario requiere editar un diagrama que se guardó, primero tiene que iniciar la sesión, dirigirse a archivo, seleccionar

abrir e iniciará el explorador de archivos con la carpeta que contiene los diagramas, selecciona el archivo el cual se abrirá en la interfaz del sistema, procediendo a editar, y tener opciones de volver a guardar o convertir el diagrama y descargar mostrado en el diagrama opción Abrir de la Figura 3.5.

Se tendrá la opción de abrir un nuevo proyecto o lienzo para hacer un diagrama nuevo, guardarlo para editar después, convertir y descargar.

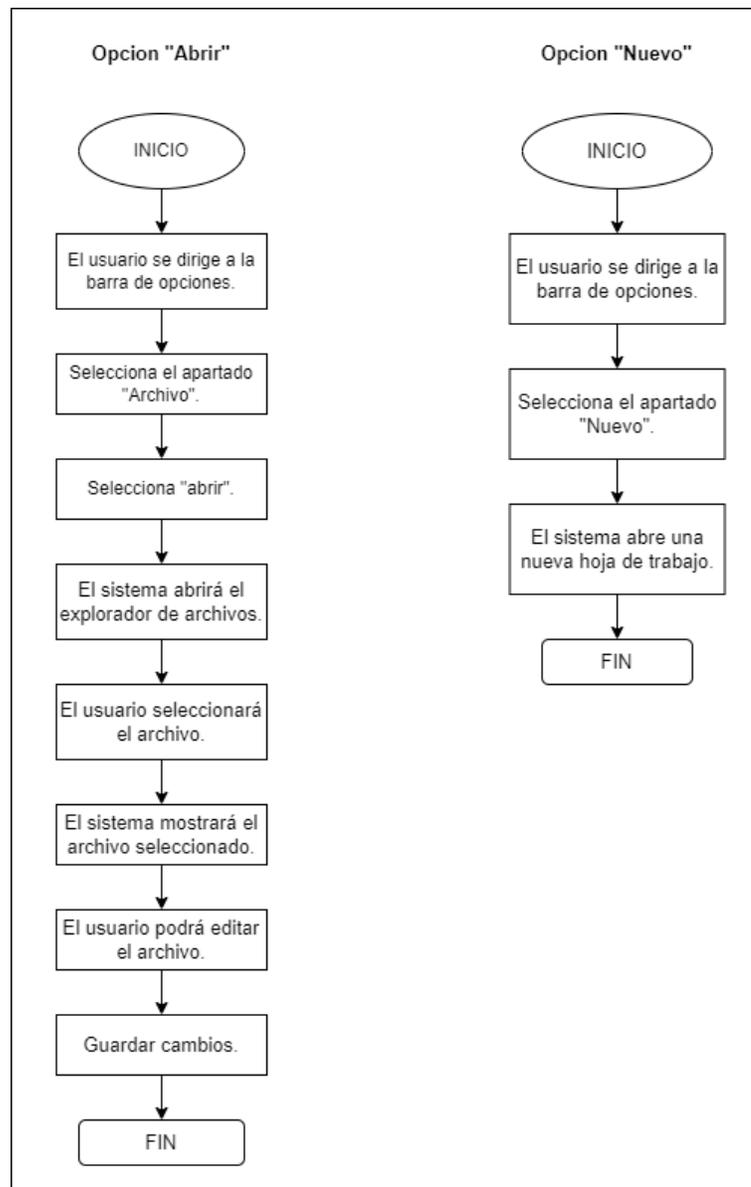


Figura 3.5: Administraci3n. Fuente: Propia

Son acciones que forman parte de las actividades fundamentales del sistema, además de Iniciar Sesión, Crear y Convertir, se presentan estas acciones que son: "Guardar como", Copiar, Pegar, Deshacer, Rehacer mostrando su función en los diagramas de las Figuras 3.6 y 3.7; los cuales los 4 últimos mencionados, son procesos básicos que llegan a ocupar los usuarios en el sistema.

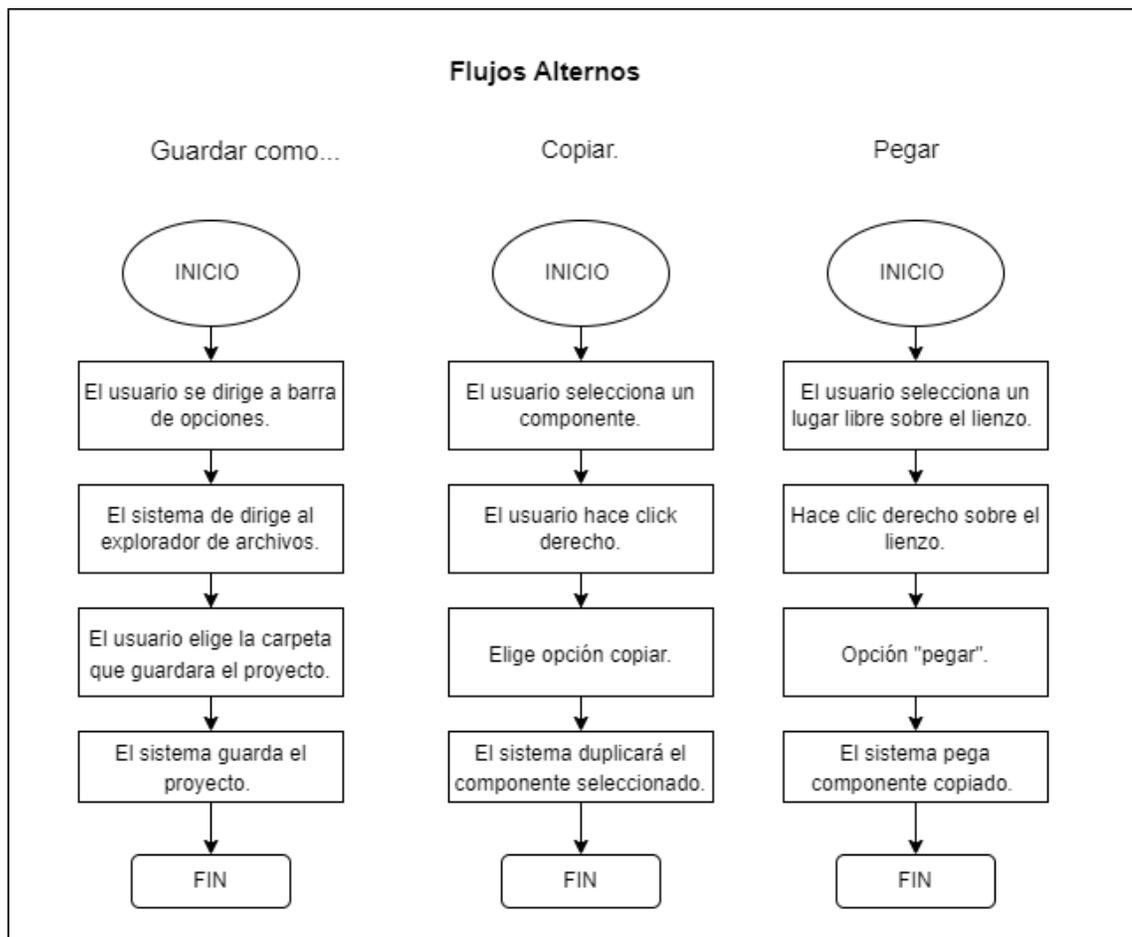


Figura 3.6: Acciones del Sistema. Fuente: Propia

En la Figura 3.7 se muestran las acciones básicas al momento de crear un diagrama, como lo son deshacer, rehacer, como forma de editar el trabajo en el sistema.

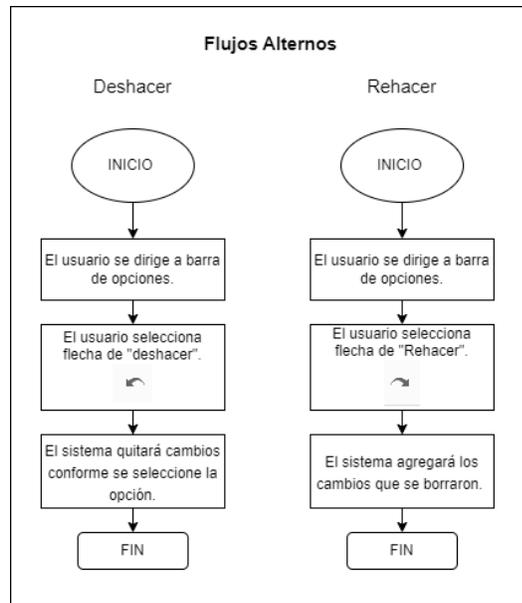


Figura 3.7: Actividades Básicas. Fuente: Propia

3.2.3 COLORES PARA APLICACIÓN

Se eligieron los colores mostrados en la Figura 3.8, como finalidad el llamar la atención del cliente y/o usuarios, además que también tienen la finalidad de ser amigable con el ojo humano, al ser un sistema que requiere de extremo uso por los usuarios al momento de crear diagramas, igual que llegan a tener afecciones emocionales, Para el ojo humano estar expuestos por un largo periodo a colores fuertes, provoca cansancio y desgaste de la vista. Los colores de la Figura 3.8, además de ser colores pastel y de tonalidades bajas a excepción del primer azul que a pesar de ser un color oscuro no llega a ser fuerte, también representan lo siguiente:

- Azul: Representa estabilidad, profundidad, transparencia, calma siendo el reflejo de valores también para el análisis y la toma de decisiones. Entrando igual como un color frío.
- Gris: Adoptando la impresión de frialdad con sensación de brillantez lujo, elegancia.

- Beige: Representa liviandad, descanso, suavidad y fluidez, dando un parecido a una cortina transparente mandando un mensaje de distención.

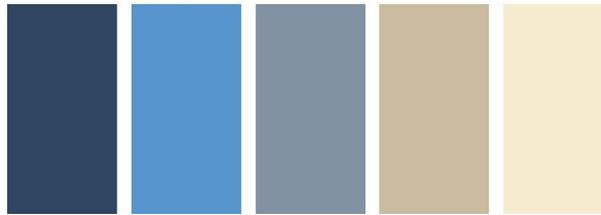


Figura 3.8: Colores para sistema. Fuente: Propia

3.2.4 MOCKUPS

Se realizaron mockups de la estructura y el acomodo de los elementos dentro el sistema. El primero que se realizó fue el mockup de Vista login/ Iniciar Sesión. Mostrado en la Figura 3.9.

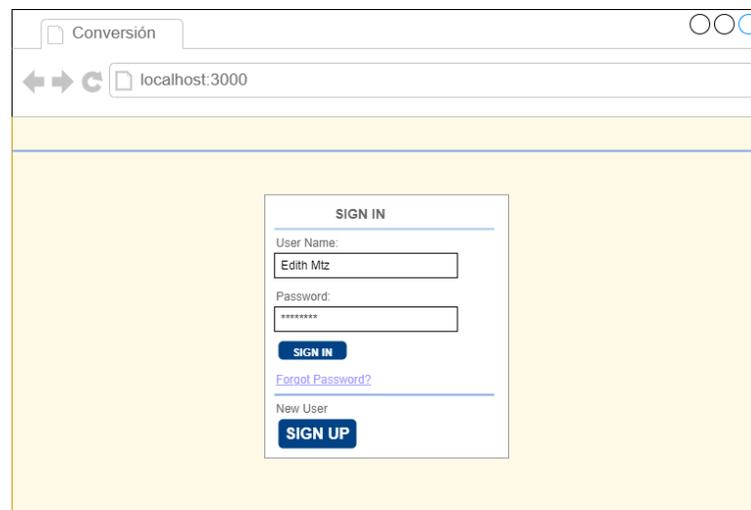


Figura 3.9: Iniciar sesión. Fuente: Propia

La interfaz principal mostrada en la Figura 3.10, presenta los elementos que el usuario podrá utilizar, mostrando que tiene 3 campos con desplegable:

1. Usuario: El desplegable de este campo cuenta con nombre de usuario y cerrar sesión.

2. Opciones: Su menú contiene apartados como Archivo, Nuevo, Guardar, "Guardar como", Abrir, Convertir.

3. Conector: Se creó un desplegable, ya que hay tipos de conector para un diagrama E-R, conteniendo puntos como: único, multivaluado, compuesto, derivado.

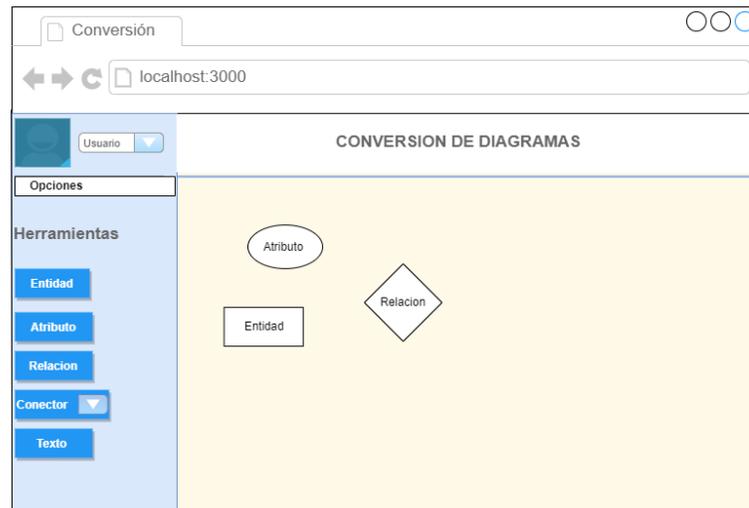


Figura 3.10: Diseño de Interfaz. Fuente: Propia

3.3 OPTIMIZACIÓN DEL DISEÑO

3.3.1 CONSTRUCCIÓN DEL PROTOTIPO

Después de evaluar y especificar el funcionamiento que tendría el sistema y a través del diseño rápido que se realizó del sistema.

Se utilizaron las aplicaciones mencionadas en el marco teórico, en las que parte de ellas son fundamentales para desarrollar el sistema.

1. De primera parte se realizó la instalación del editor de texto, es decir, Visual Studio Code, como su función lo dice al ser un editor de texto y contar con una terminal, se pudo crear las pantallas, funciones, detalles y texto que requirió el sistema.

2. Se realizó la instalación de Node.js siendo un entorno de código abierto en el que se ejecutará el código de JavaScript, al igual que servirá como servidor, tomemos en cuenta que se está utilizando la versión de Node.js 18.7 y la versión de npm 8.15 del entorno, como se muestra en la Figura 3.11.

```
C:\Users\mared>node -v
v18.7.0

C:\Users\mared>npm -v
8.15.0

C:\Users\mared>_
```

Figura 3.11: Versiones node. Fuente: Propia

3. Se realizó la instalación dentro de Visual Studio Code las librerías necesarias del lienzo de dibujo Canvas y las necesarias de React.js, que son los formatos en los que se trabajará para el desarrollo del sistema.

4. Como primera vista se obtuvo el siguiente resultado, Figura 3.12, con los colores especificados anteriormente, se tomó en cuenta que en esta primera imagen se dá a conocer el acomodo de los colores.

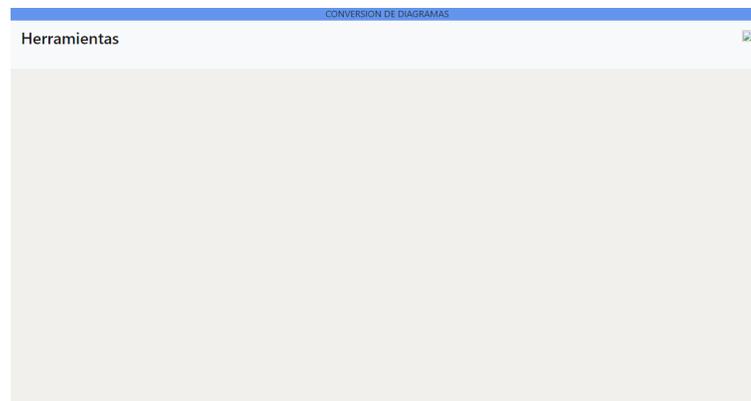


Figura 3.12: Vista Práctica. Fuente: Propia

5. Función de la terminal de Visual Studio Code y Node.js para encender el servidor de este último, la ventaja de este programa es que, al momento de guardar cambios, en la terminal se estará actualizando mostrando ‘web pack Compiled Successfully’ como se muestra en la Figura 3.13.

```
PS C:\Users\mared\OneDrive\Escritorio\my-app> npm start

> my-app@0.1.0 start
Compiled successfully!
Compiled successfully!

You can now view my-app in the browser.

  Local:          http://localhost:3000
  On Your Network: http://192.168.0.7:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
█
```

Figura 3.13: Terminal Npm Iniciada. Fuente: Propia

6. Se realizó la investigación de un formato y el código para mostrar una barra vertical de lado izquierdo de la pantalla, de la cual se utilizó un comando «Sidebar», para especificar y mostrar el apartado especial para los botones de las herramientas; cabe recalcar que al utilizar React.js la estructura y organización del código dentro de este entorno se maneja por componentes, por lo que «Sidebar» es uno de los componentes que se utilizaron.

7. Se especificaron los demás componentes como «Navbar» siendo una barra horizontal normal para título y logo, dentro de los componentes se registraron por separado los botones «botton» para los elementos de diagramas E-R.

3.4 EVALUACIÓN DEL DISEÑO Y PROTOTIPO

3.4.1 EVALUACIÓN DEL PROTOTIPO POR EL CLIENTE

Se realizaron dos pruebas en el sistema, en estas pruebas se abarco la parte visual mostrada en la Tabla 3.1, con la finalidad de mostrar la interfaz completa del sistema, mostrando sus elementos, y herramientas de trabajo.

La prueba 1 se basó en verificar funcionamiento visual, incluyendo que el local host de React muestre el proyecto, el diseño gráfico, lienzo y apartados completos.

Tabla 3.1: Prueba 1.

NOMBRE DE LA PRUEBA	PRUEBA 1
Objetivo de la prueba	Mostrar correctamente la interfaz principal con todos los componentes.
Descripcion de la prueba	Se debe de ver la interfaz estática con todos los elementos que debe de tener, es decir, el lienzo y barra vertical.
Condiciones de la prueba	Corre o ejecuta correctamente el proyecto.
Resultados esperados	Se muestra los diferentes apartados mencionados y desglosa la función de respuesta de los menús con desglose.
Resultados esperados	Los apartados se muestran correctamente.

Por otra parte, la siguiente prueba de la Tabla 3.2, se realizó para verificar la función de los botones mostrados en la barra vertical creada, ubicada en la parte izquierda del sistema, se evalúa que en la mayoría de los botones se mostrara la función correcta, al arrojar el elemento que se seleccionó.

Tabla 3.2: Prueba 2.

NOMBRE DE LA PRUEBA	PRUEBA 2
Objetivo de la prueba	Mostrar la funcionalidad de ver la interfaz con la interacción de los componentes del modelo E-R.
Descripcion de la prueba	Verificación de funcionalidad, interacción de los componentes del modelo E-R, selección, arrojar y editar en la hoja de trabajo.
Condiciones de la prueba	Dar clic en algún componente del modelo E-R.
Resultados esperados	Que el sistema muestre el componente seleccionado en la hoja de trabajo.
Resultados actuales	El sistema mostró la mayoría de componentes seleccionados.

3.4.2 REFINAMIENTO DEL PROTOTIPO.

- Recolección de errores.
- Se detectaron una mezcla de errores desde la instalación de las herramientas que se utilizaron para el desarrollo del sistema.
- Se detectó problemas de función con la librería Canvas.
- Realización de correcciones correspondientes a los errores surgidos.
- Detectó la situación por incompatibilidad de versiones, realizando nuevamente la instalación de otras versiones actuales.

- Detectó la situación por incompatibilidad de versiones, realizando nuevamente la instalación de otras versiones actuales.
- Se analizó código, se detectó falta de código para llamado de Funciones de la librería.
- Implementación de mejoras.
- Integración de posibles cambios para mejorar el sistema.
- Análisis de nuevas funciones para el sistema que benefician el usuario.

3.5 CRONOGRAMA

En la Figura 3.14 se muestra las actividades que se abarcaron a lo largo del periodo de desarrollo de las residencias.

Actividades	Agosto	Septiembre	Octubre	Noviembre	Diciembre
Obtención de requerimientos a través de un CRS.					
Diseños las vistas del sistema.					
Definición del funcionamiento del sistema.					
Desarrollo del funcionamiento del sistema.					
Pruebas de funcionamiento.					
Recolección de errores.					
Realización de modificaciones y mejoras.					
Realizar prueba de funcionamiento.					
Sistema terminado.					

Figura 3.14: Cronograma. Fuente: Propia

CAPÍTULO 4

RESULTADOS

4.1 RESULTADOS OBTENIDOS

4.1.1 PRODUCTO DE INGENIERÍA

Parte presentada como la interfaz principal del sistema, muestra los colores seleccionados visualizados en la Figura 4.1:



Figura 4.1: Interfaz Principal. Fuente: Propia

Se realizó la ejecución de los botones para verificar el funcionamiento de los mismos,

al igual que el funcionamiento del lienzo, es decir, que el mismo dejara libertad de movimiento de las figuras.

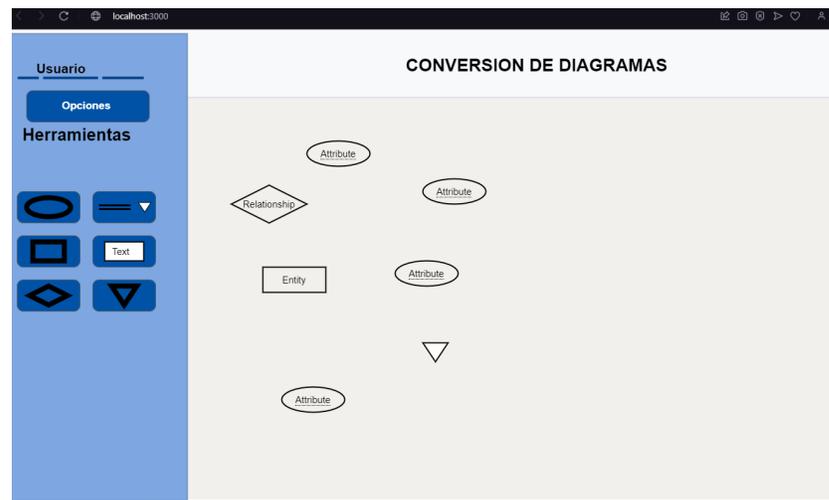


Figura 4.2: Botones y lienzo. Fuente: Propia

A continuación en la Figura 4.2 se muestra como el sistema quedo hasta el momento, mostrando la interfaz principal y el área del lienzo para que el usuario pueda, crear, editar, mover y agregar. Muestra la ejecución de los botines de entidad, texto, conector, atributos, relación y generalización.

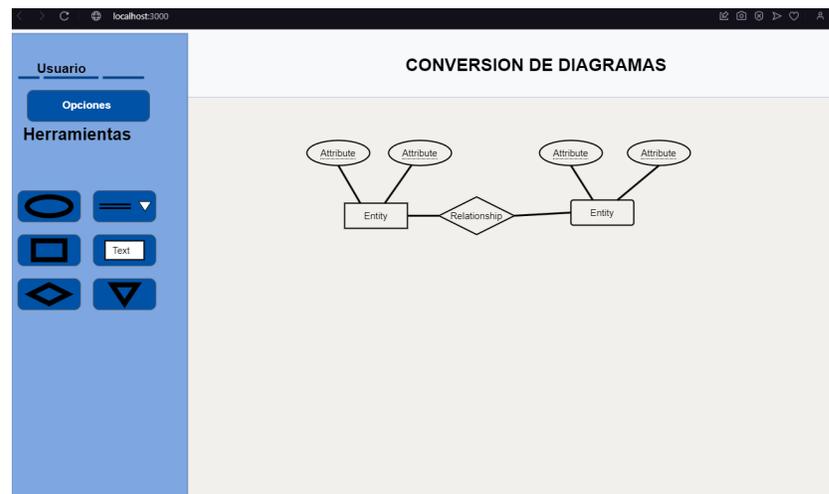


Figura 4.3: Diagrama terminado. Fuente: Propia

En el sistema se muestra como los elementos, en este caso, las figuras que representan las entidades, atributos, relaciones, conectores u otros componentes

pueden moverse alrededor del lienzo, haciendo una comparativa entre la Figura 4.2 y la Figura 4.3, como los componentes se movieron en diferente área, sin ninguna complicación.

Las figuras están condicionadas para que muestren un título predeterminado, su propósito es que el usuario tenga una guía en que parte puede escribir de la figura, esto incluye en el componente "Text" que al ser de tipo texto y no tiene un marco como las figuras puede llegar a perderse para el usuario.

CAPÍTULO 5

CONCLUSIONES

5.1 CONCLUSIONES DEL PROYECTO

Lo mostrado a lo largo de este trabajo permitió llegar a varias conclusiones. Se analizó la importancia del modelado y conversión de modelos E-R a relacional, se obtuvo conocimiento de la importancia de los sistemas de información en general. En el ámbito de la informática se tenía como prioridad el adquirir una herramienta más completa al momento de modelar y convertir modelos E-R a relacional, la cual ayudaría a profesionales como a estudiantes y profesores. Se tomó en cuenta de la importancia que debe de tener la teoría de las bases de datos y del porqué.

Se desarrollaron métodos de investigación para la obtención de los requerimientos y obtener información para proceso de conversión. Se obtuvo conocimiento por medio de cursos para el correcto desarrollo de actividades y funcionalidades del sistema en los entornos JavaScript, React.js, Canvas y Node.js

Se desarrolló una herramienta que nos permite todas las cosas mencionadas anteriormente, investigamos herramientas existentes que emulaban o hacían incompleto el trabajo, con esta herramienta se obtuvo el análisis riguroso, el modelado, investigación del proceso de conversión y el seguimiento de la teoría de la base de datos.

CAPÍTULO 6

COMPETENCIAS DESARROLLADAS

6.1 COMPETENCIAS DESARROLLADAS Y/O APLICADAS

- 1. Apliqué habilidades directivas y de ingeniería en el diseño, gestión, fortalecimiento e innovación de las organizaciones para la toma de decisiones en forma efectiva, con una orientación sistémica y sustentable.
- 2. Diseñé e Innové estructuras administrativas y procesos tecnológicos, con base en las necesidades de las organizaciones para competir.
- 3. Apliqué métodos cuantitativos y cualitativos en el análisis e interpretación de datos y modelado de sistemas y modelado de datos en diagramas relacionales en los procesos organizacionales.
- 4. Diseñé y emprendí proyectos empresariales sustentables en mercados competitivos, para promover el desarrollo tecnológico.
- 5. Diseñé e implemente estrategias de mercadotecnia basadas en información recopilada de fuentes primarias y secundarias, para incrementar la competitividad de las organizaciones.
- 6. Gestioné sistemas integrales de calidad para la mejora de los procesos, ejerciendo un liderazgo estratégico y un compromiso ético. etc.

APÉNDICE A

ANEXOS

A.1 ANEXOS

(carta de autorización por parte de la empresa u organización para la residencia profesional y otros si son necesarios).

A.2 REGISTROS DE PRODUCTOS O MARCAS

(patentes, derechos de autor, compra-venta del proyecto, etc.)

BIBLIOGRAFÍA

- [1] Ayudaley, “El modelo base de datos: Definición y tipos.”
- [2] BD, “Elaboración de modelos relacionales de bases de datos,” 2022.
- [3] L. A. H. Gonzalez, “Modelo e-r extendido (ee-r),” 2013.
- [4] PowerDesigner, “Funcionalidades principales de powerdesigner,” 2022.
- [5] M. L. A. Latukolan, A. Arwan, and M. T. Ananta, “Pengembangan sistem pemetaan otomatis entity relationship diagram ke dalam database,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN*, vol. 2548, p. 964X, 2019.
- [6] Z. Y. Cama, “Desarrollo de una herramienta case para el diseño de diagramas entidad–relación extendido y su mapeo al modelo relacional orientado a estudiantes en el contexto,” *INF-FCPN-PGI Revista PGI*, pp. 210–213, 2021.
- [7] A. Barreto Sánchez and J. Fuentes Gallardo, “Plugin para generar esquemas dimensionales a partir del diagrama entidad-relación desde el visual paradigm,” B.S. thesis, Universidad de las Ciencias Informáticas. Facultad-6., 2016.
- [8] A. E. Díaz *et al.*, “Sitio web “aprender modelo entidad-relación”,” *Revista Conrado*, vol. 12, no. 56, 2016.
- [9] incap, “Sistema de información,” s.f.
- [10] Microsoft, “Conceptor basicos sobre bases de datos,” s.f.

-
- [11] A. Silberschatz, “Fundamentos de base de datos,”
file:///C:/Users/mared/Dropbox/Mis.f.
- [12] *OpenWebinars.net*, “Qué es nodejs y para qué sirve.,” 2019.
- [13] *Openwebinars*, “Qué es visual studio code y qué ventajas ofrece,” 2022.
- [14] *Developer*, “Api canvas,” s.f.
- [15] *React.js*, “React — qué es, para qué sirve y cómo funciona — descúbrelo todo,” 2021.
- [16] *Developer*, “Fundamentos de javascript,” 2022.
- [17] M. J. Guapi Auquilla, “Diseño metodológico para el desarrollo de interfaces gráficas en páginas web utilizando los lenguajes html 5 y css 3,” *B.S. thesis, Riobamba, 2018.*
- [18] S. Atut and D. Howell, “Analysis of case-systems in business planning,” *International Journal of Millennium Studies*, vol. 6, no. 3, 2021.