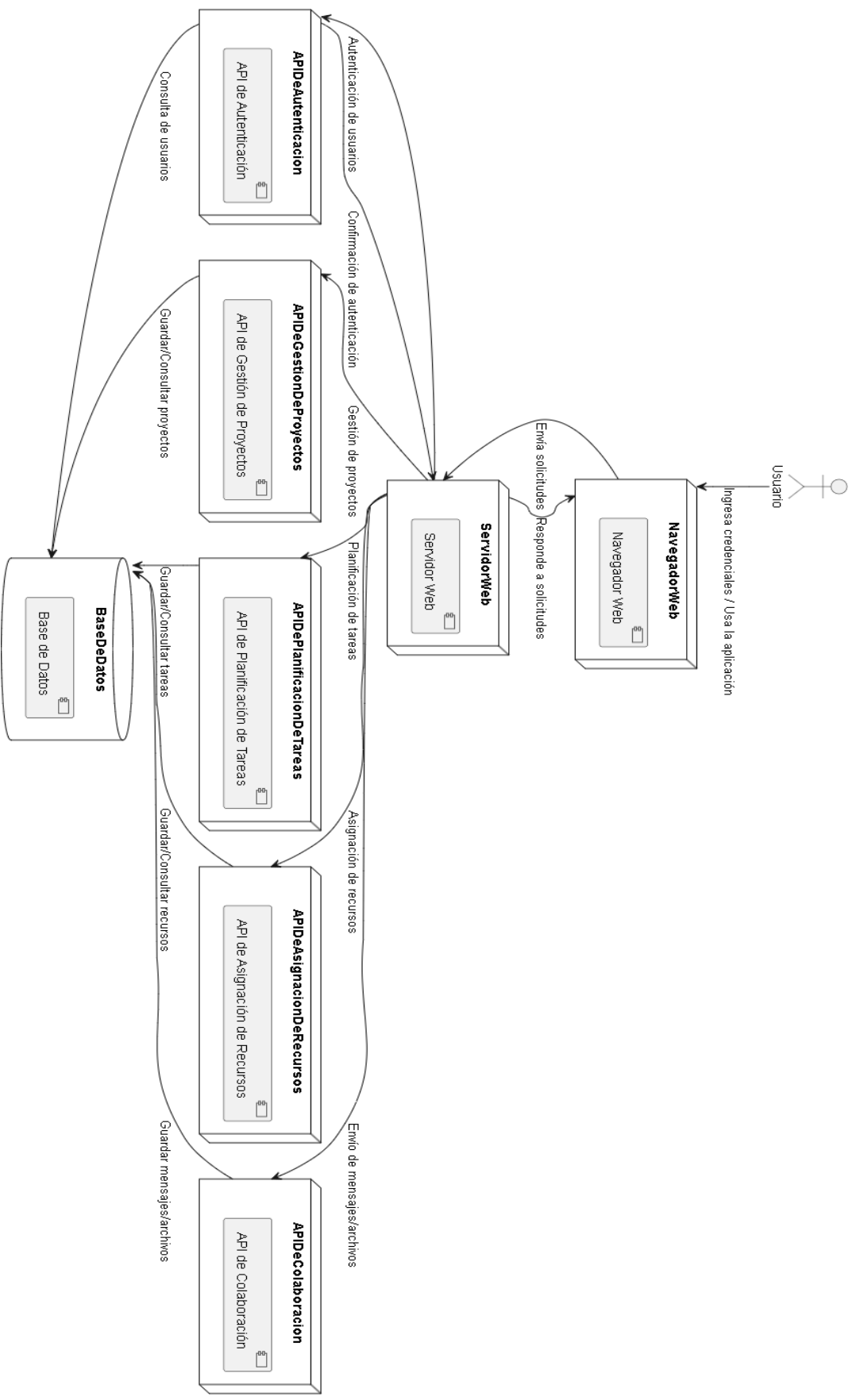


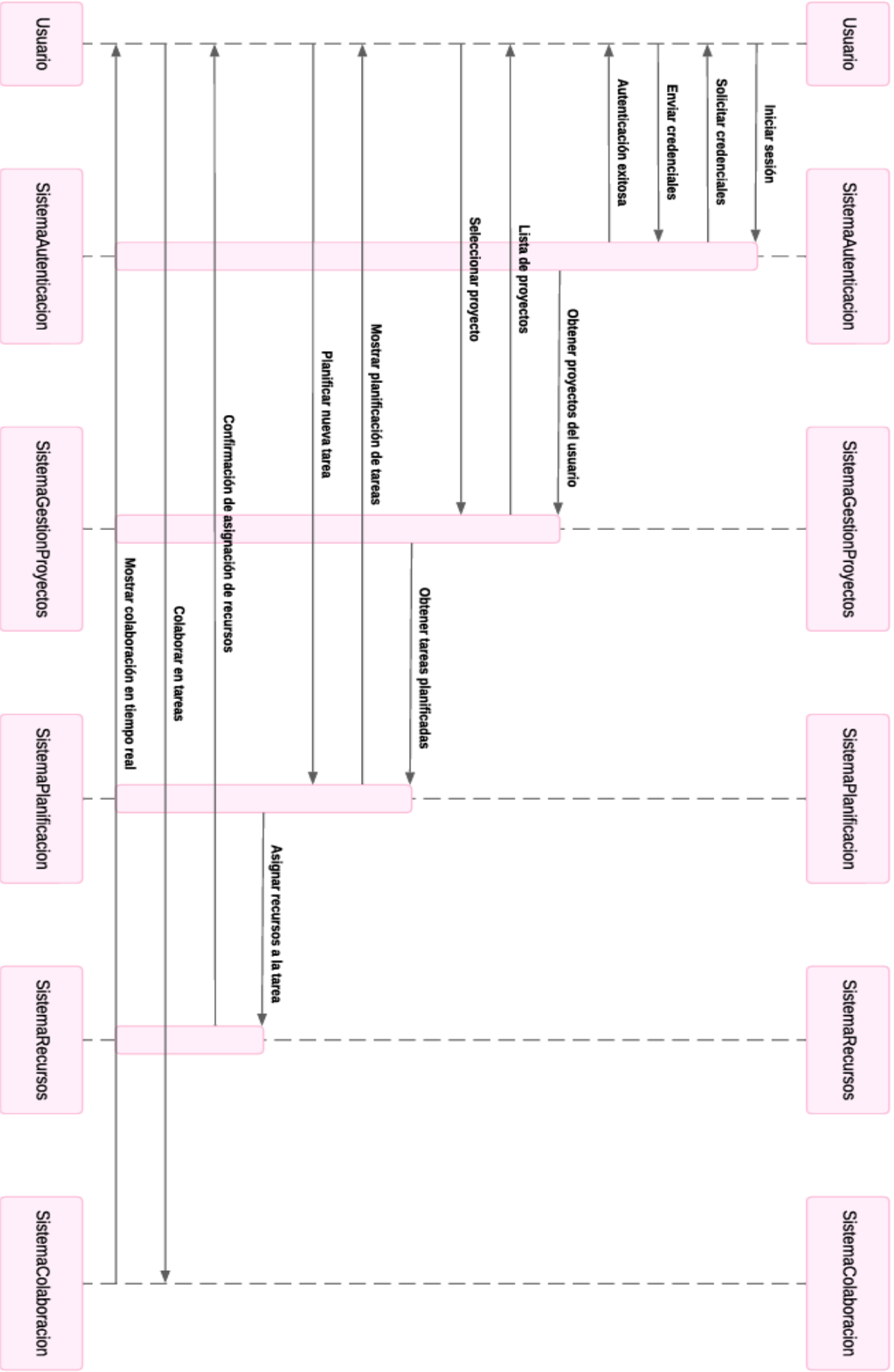
1.10 DIAGRAMA DE DISTRIBUCION

Diagrama 21.



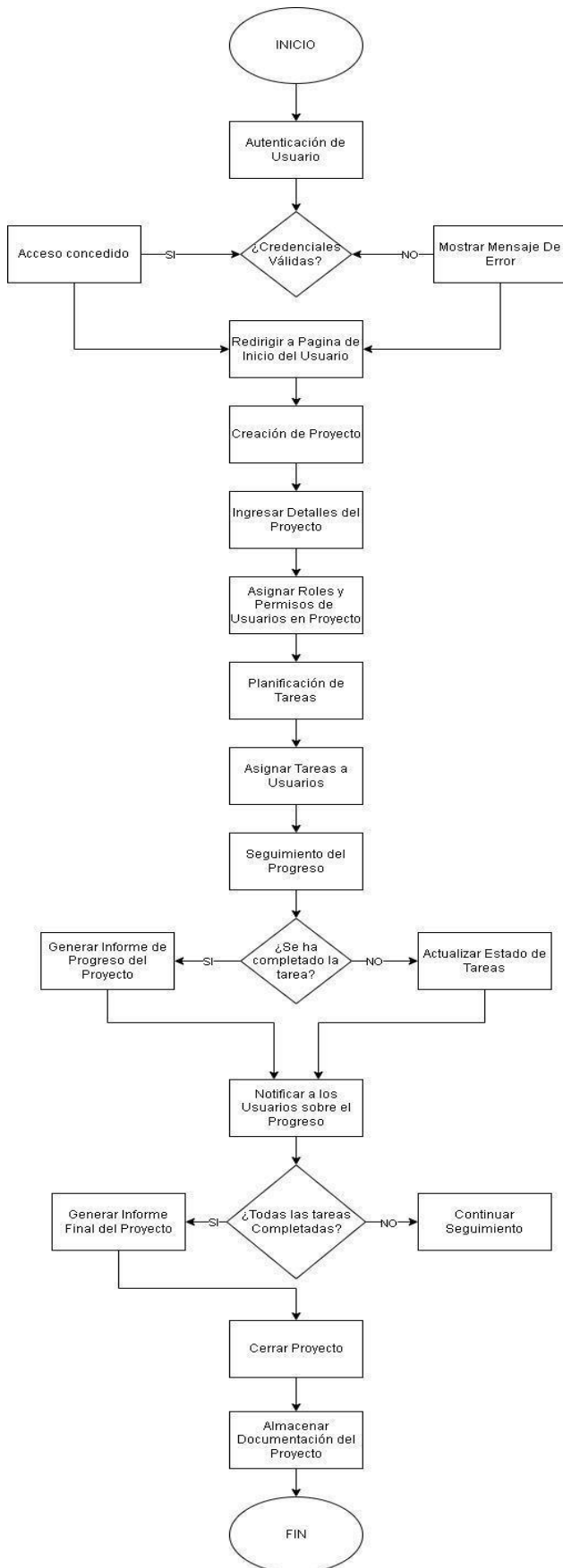
1.11 DIAGRAMA DE SECUENCIA

Diagrama 22.



1.12 DIAGRAMA DE FLUJO

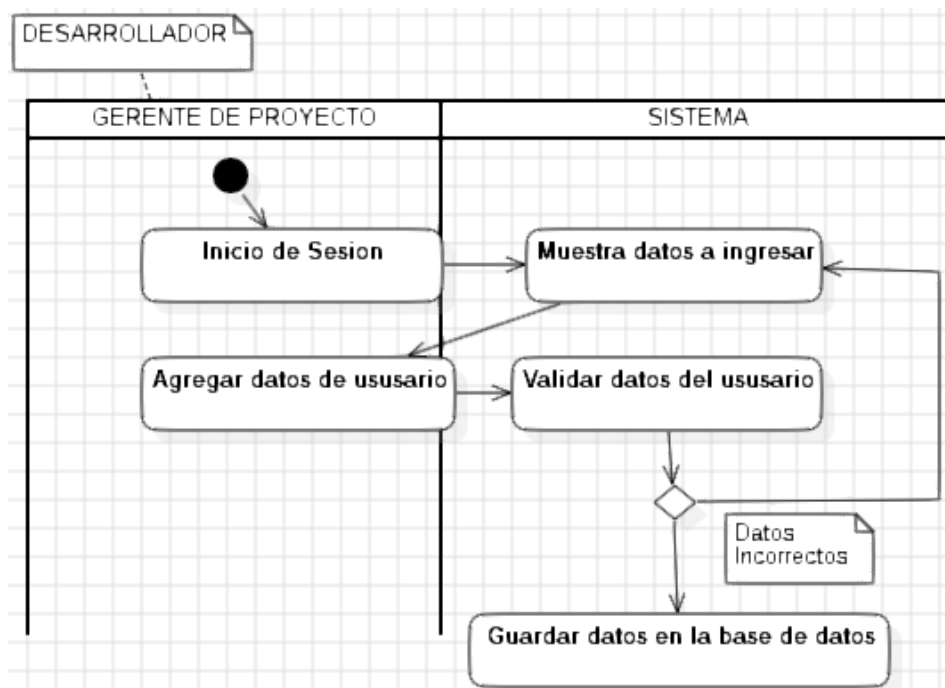
Diagrama 23.



1.13 DIAGRAMA DE ACTIVIDADES

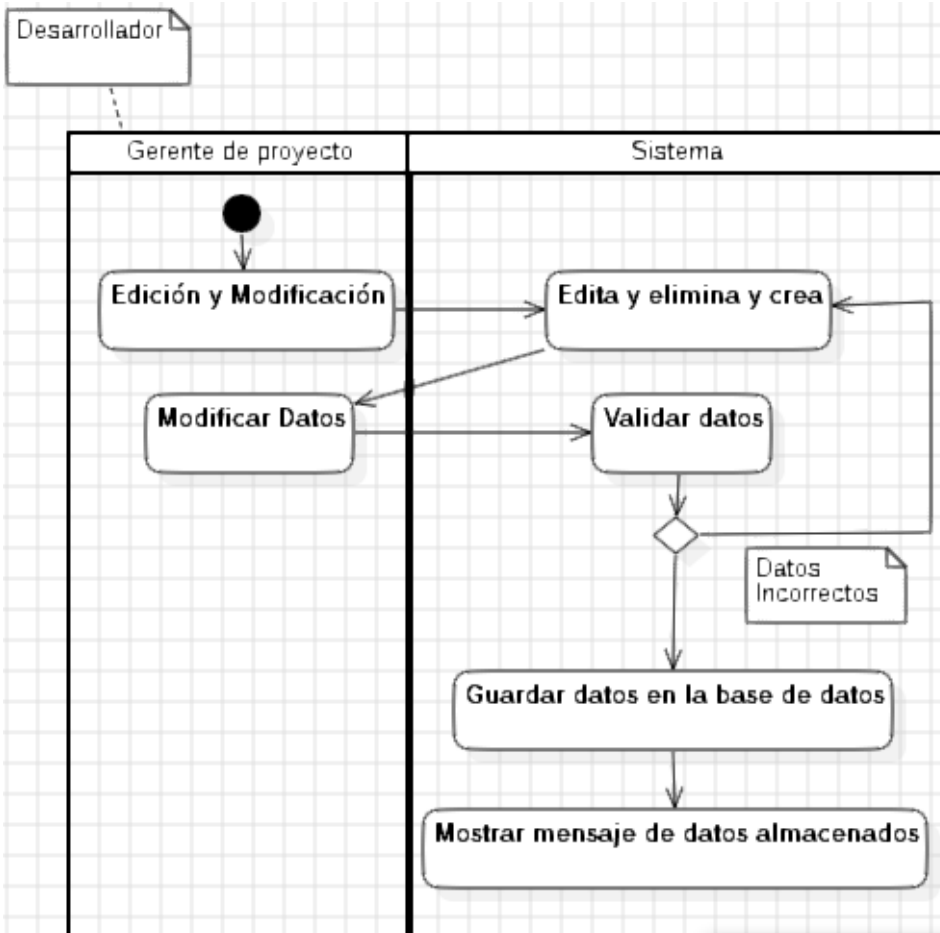
Inicio Sesión:

Diagrama 25.



Edición y modificación:

Diagrama 26.



1.14 EXPLICACION DEL MODELO DE LA BD, TABLAS Y ESTRUCTURA

El modelo de base de datos (BD) para este proyecto se basa en una arquitectura relacional que permita gestionar de manera eficiente los proyectos, las tareas y la relación entre usuarios, roles, y permisos. Las tablas clave incluyen:

Usuarios: Guarda información de los usuarios del sistema, como gerentes y desarrolladores.

Proyectos: Contiene los detalles de los proyectos, como el nombre, descripción y fechas.

Tareas: Almacena las tareas de los proyectos, asignadas a los usuarios.

Roles: Define los roles dentro del sistema, como gerente de proyecto o desarrollador.

Permisos: Gestiona los permisos que se otorgan a los usuarios de acuerdo con sus roles.

Historial de Actividades: Registra todas las acciones que los usuarios realizan en los proyectos para tener un seguimiento completo.

Cada tabla tiene relaciones de cardinalidad específicas para modelar correctamente las interacciones y el flujo de información entre usuarios, proyectos y tareas.

Estructura y campos clave:

Usuarios (ID Usuario, Nombre, Correo, Contraseña, ID Rol)

Proyectos (ID Proyecto, Nombre Proyecto, Descripción, Fecha Inicio, Fecha Fin, ID Gerente)

Tareas (ID Tarea, Nombre Tarea, Descripción Tarea, Estado, ID Usuario, ID Proyecto, Fecha Asignación, Fecha Vencimiento)

Roles (ID Rol, Nombre Rol)

Permisos (ID Permiso, Nombre Permiso, ID Rol)

Historial de Actividades (ID Historial, ID Usuario, ID Tarea, Acción, Fecha Acción)

Deficiencias de información del proyecto para generar la estructura y la BD

Para completar el diseño de la base de datos, faltan los siguientes detalles:

Requisitos adicionales: No se han definido campos específicos para informes, notificaciones y automatización de procesos, lo cual influye en la estructura de la BD.

Relaciones detalladas: Se necesita mayor claridad sobre cómo se relacionan las tareas entre proyectos y si se pueden compartir tareas entre proyectos.

Gestión de recursos: No está claro si el sistema debe administrar recursos financieros o materiales, lo cual podría requerir tablas adicionales.

Cardinalidad

Usuarios a Roles: Muchos a uno. Un usuario tiene un solo rol, pero un rol puede estar asociado a muchos usuarios.

Proyectos a Usuarios (Gerentes): Muchos a uno. Cada proyecto tiene un solo gerente, pero un gerente puede gestionar varios proyectos.

Proyectos a Tareas: Uno a muchos. Cada proyecto puede tener varias tareas.

Tareas a Usuarios (Asignados): Muchos a uno. Cada tarea es asignada a un solo usuario, pero un usuario puede tener varias tareas.

Permisos a Roles: Muchos a uno. Un rol puede tener múltiples permisos, pero un permiso puede aplicarse a varios roles.

Diagramas de BD

1. Modelo Conceptual:

Incluye las entidades principales (Usuarios, Proyectos, Tareas, Roles, Permisos, Historial) y sus relaciones.

MODELO CONCEPTUAL

USUARIOS
+id
+Nombre
+Email
+Contraseña
+Fecha de Creación

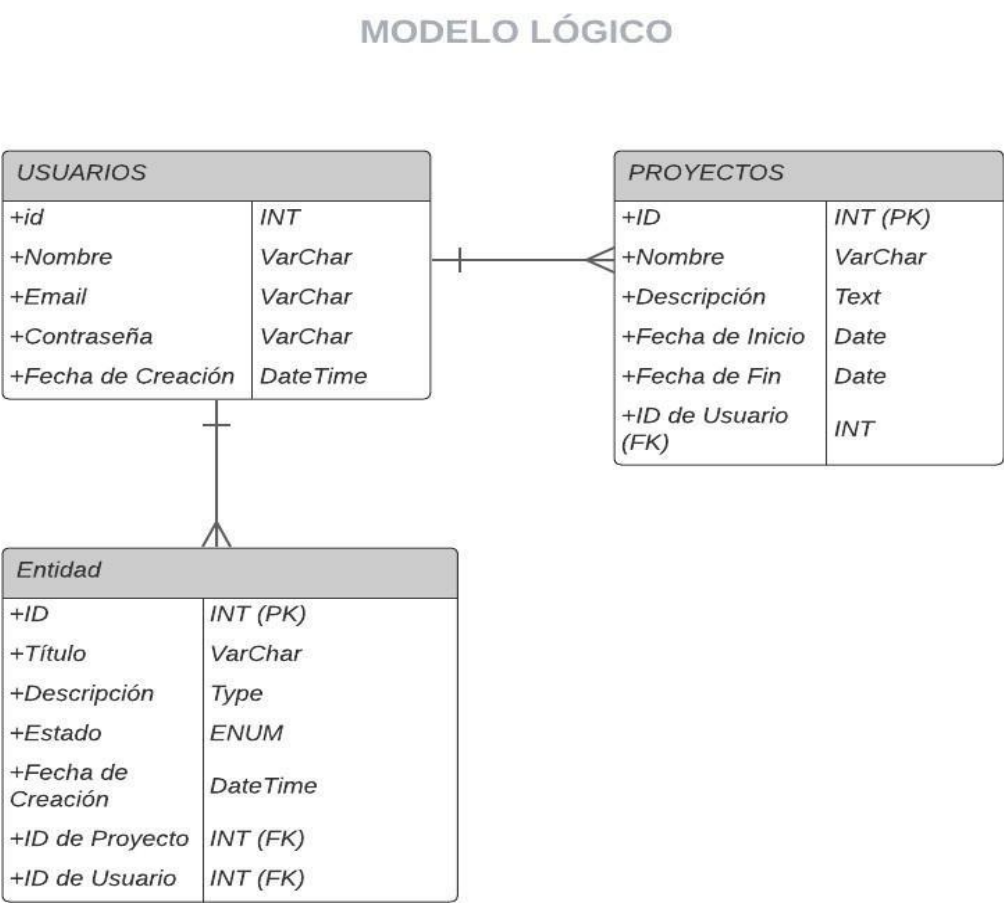
Entidad
+ID
+Título
+Descripción
+Estado
+Fecha de Creación
+ID de Proyecto
+ID de Usuario

PROYECTOS
+ID
+Nombre
+Descripción
+Fecha de Inicio
+Fecha de Fin
+ID de Usuario (FK)

2. Modelo Lógico:

Representa las relaciones y los campos principales en cada entidad.

Relaciones entre las tabla

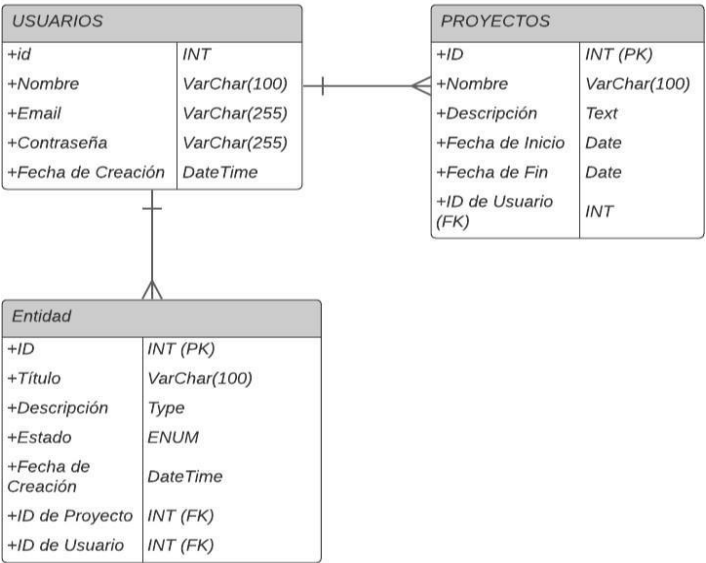


s, tipos de datos, claves primarias y claves foránea

3. Modelo Físico:

Define las tablas con todos los detalles técnicos: tipos de datos específicos (VARCHAR, DATE, etc.), restricciones de integridad, índices, y relaciones entre las

MODELO FÍSICO



tablas a nivel físico.

Script SQL de la base de datos del proyecto:

-- Creación de la base de datos

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

-- Tabla: Usuarios

```
CREATE TABLE IF NOT EXISTS Usuarios (  
ID INT AUTO_INCREMENT PRIMARY KEY,  
Nombre VARCHAR(100) NOT NULL,  
Email VARCHAR(255) NOT NULL UNIQUE,  
Contraseña VARCHAR(255) NOT NULL,  
FechaCreacion DATETIME DEFAULT CURRENT_TIMESTAMP  
);
```

-- Tabla: Proyectos

```
CREATE TABLE IF NOT EXISTS Proyectos (  
ID INT AUTO_INCREMENT PRIMARY KEY,  
Nombre VARCHAR(100) NOT NULL,  
Descripcion TEXT,  
FechaInicio DATE NOT NULL,  
FechaFin DATE NOT NULL,  
IDUsuario INT,  
FOREIGN KEY (IDUsuario) REFERENCES Usuarios(ID) ON  
DELETE CASCADE  
);
```

```
-- Tabla: Tareas

CREATE TABLE IF NOT EXISTS Tareas (

ID INT AUTO_INCREMENT PRIMARY KEY,

Titulo VARCHAR(100) NOT NULL,

Descripcion TEXT,

Estado ENUM('Pendiente', 'En Progreso', 'Completo') NOT NULL,

FechaCreacion DATETIME DEFAULT CURRENT_TIMESTAMP,

IDProyecto INT,

IDUsuario INT,

FOREIGN KEY (IDProyecto) REFERENCES Proyectos(ID) ON

DELETE CASCADE,

FOREIGN KEY (IDUsuario) REFERENCES Usuarios(ID) ON

DELETE CASCADE

);
```

Explicación del Script:

Creación de la base de datos: El script inicia creando la base de datos ProyectoGestionsi no existe y luego selecciona esa base de datos para realizar las siguientes operaciones.

Tabla Usuarios:

ID: Identificador único de cada usuario, auto-incremental.

Nombre, Email, Contraseña: Campos para almacenar la información del usuario.

FechaCreacion: Se establece automáticamente en la fecha y hora actual es al momento de la creación del registro.

Tabla Proyectos:

ID: Identificador único del proyecto.

Nombre, Descripción, FechaInicio, FechaFin: Campos que almacenan los detalles del proyecto.

IDUsuario: Clave foránea que hace referencia al ID de la tabla Usuarios.

Tabla Tareas:

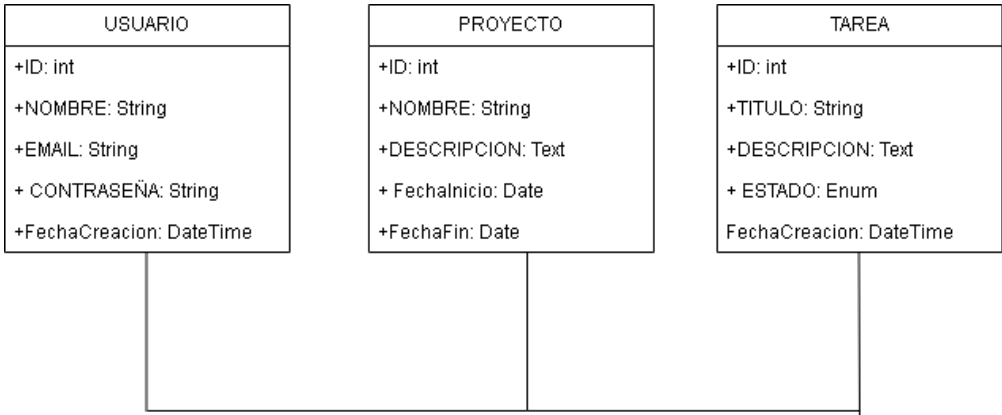
ID: Identificador único de la tarea.

Título, Descripción, Estado: Campos que almacenan los detalles de la tarea.

FechaCreacion: Se establece automáticamente en la fecha y hora actuales.

IDProyecto, IDUsuario: Claves foráneas que hacen referencia a las tablas Proyectos y Usuarios, respectivamente.

1.15 MAPA DE NAVEGACION



Gerente de Proyecto

Pantalla de Inicio de Sesión

Login exitoso → **Dashboard Principal**

Dashboard Principal

Ver listado de proyectos → **Vista de Proyectos**

Crear nuevo proyecto → **Formulario de Creación deProyecto**

Revisar notificaciones → **Panel de Notificaciones**

Vista de Proyectos

Seleccionar proyecto → **Vista de Detalle del Proyecto**

Editar/Eliminar proyecto → **Formulario de Edición deProyecto**

Vista de Detalle del Proyecto

Ver tareas del proyecto → **Vista de Tareas**

Asignar usuarios a tareas → **Asignación de Tareas**

Vista de Tareas

Crear nueva tarea → **Formulario de Creación de Tarea**

Editar/Eliminar tarea → **Formulario de Edición de Tarea**

Usuario Regular

Pantalla de Inicio de Sesión

Login exitoso → **Dashboard Principal**

Dashboard Principal

Ver tareas asignadas → **Vista de Tareas**

Actualizar estado de tarea → **Edición de Tarea**

Vista de Tareas

Detalles de la tarea → **Vista de Detalle de Tarea**

Actualizar estado de tarea → **Formulario de Actualización**

Lenguajes Utilizados y Justificación

HTML (HyperText Markup Language):

Descripción: HTML es el lenguaje base del web utilizado para estructurar el contenido de las páginas web.

Justificación: HTML permite estructurar de manera clara el contenido, con secciones definidas para elementos como el inicio de sesión, el registro, la lista de tareas, y otras características del proyecto. Su simplicidad, compatibilidad y estandarización en todos los navegadores lo hacen esencial.

CSS (Cascading Style Sheets):

Descripción: CSS permite definir el diseño y la apariencia visual de los elementos de HTML.

Justificación: CSS fue seleccionado para brindar una interfaz visualmente atractiva y profesional que se ajuste a una experiencia de

usuario intuitiva. Además, CSS permite la adaptación de la página en diferentes dispositivos, proporcionando una experiencia de diseño responsivo.

JavaScript

Descripción: JavaScript es un lenguaje de programación que permite crear interactividad y dinamismo en la web.

Justificación: Es un lenguaje clave para el desarrollo de aplicaciones web dinámicas y se utiliza ampliamente para gestionar interacciones en la interfaz de usuario. En este proyecto, JavaScript será fundamental para validar formularios, manejar el flujo de la interfaz y permitir acciones como la creación y modificación de tareas sin necesidad de recargar la página.

SQL (Structured Query Language):

Descripción: SQL es el lenguaje estándar para el manejo de bases de datos relacionales.

Justificación: SQL permite gestionar de manera eficiente la base de datos del sistema, que contiene la información de los usuarios, proyectos y tareas. Su capacidad de realizar operaciones de consulta complejas, junto con su fiabilidad y velocidad en entornos de producción, lo hace la elección más adecuada para esta plataforma.

Estructura del Proyecto

Frontend (Interfaz de usuario):

Construido con **HTML** y **CSS**, permite a los usuarios interactuar con el sistema y realizar acciones de forma intuitiva. **JavaScript** se usa

aquí para controlar la lógica en el cliente, manejando eventos como el inicio de sesión y la creación de nuevas tareas.

Backend y Base de Datos:

El backend emplea **SQL** para las operaciones de base de datos. Las interacciones entre la interfaz y la base de datos se manejan de manera segura para garantizar que los datos de los usuarios y las tareas se procesen y almacenen de forma eficiente.

