

EVALUACION SUMATIVA 2

Taller de Diseño de Sistemas

**MÓDULO: ISI601**

**DOCENTE: JUAN DONOSO**

**ESTUDIANTE: CRISTINA FARIÑO PÉREZ**

Índice

[Resumen 3](#_Toc197982940)

[Introducción 3](#_Toc197982941)

[Desarrollo 4](#_Toc197982942)

[Normalización de la Base de Datos 4](#_Toc197982943)

[**Primera Forma Normal (1FN)** 4](#_Toc197982944)

[**Segunda Forma Normal (2FN)** 4](#_Toc197982945)

[Diagramas Lógico y Físico 5](#_Toc197982946)

[**Diagrama Lógico** 5](#_Toc197982947)

[**Diagrama Físico** 6](#_Toc197982948)

[Sistema 10](#_Toc197982949)

[Instrucciones de uso 10](#_Toc197982950)

[- **Inicio del Sistema:** 10](#_Toc197982951)

[Conclusión 14](#_Toc197982952)

[Referencias bibliográficas 15](#_Toc197982953)

# **Resumen**

Este informe presenta el proceso de normalización de una base de datos de la sucursal de Scranton de Dunder Mifflin. El objetivo es transformar una estructura de datos ineficiente en un modelo relacional que funcione como base para un sistema ERP. Se aplicaron la Primera y Segunda Forma Normal (1FN y 2FN) para dividir la tabla original en entidades más simples, eliminando redundancias y mejorando la integridad de los datos. También se incluyen diagramas lógico y físico, el script SQL para crear la base de datos en MySQL, una descripción de los CRUDs desarrollados y el repositorio del proyecto en GitHub.

# **Introducción**

Dunder Mifflin es una empresa ficticia de distribución de papel, conocida por su estilo de trabajo único. En la sucursal de Scranton, toda la información administrativa se guarda en un archivo Excel en un disquete. Debido a los riesgos que esto conlleva, se ha visto la necesidad de migrar a un sistema más moderno, por lo que se propone construir un ERP mínimo viable (MVP) basado en una base de datos normalizada. Este documento detalla el proceso de creación de esta base, su estructura lógica y física, el script de implementación y los formularios para la gestión de datos (CRUDs), todo documentado.

# **Desarrollo**

## **Normalización de la Base de Datos**

### **Primera Forma Normal (1FN)**

La Primera Forma Normal (1FN) establece que todos los atributos de una tabla deben contener valores atómicos, es decir, no deben existir listas ni conjuntos de valores dentro de una sola celda. Además, las filas deben ser únicas y no debe haber duplicación de columnas. En el caso de la hoja original extraída del archivo Excel de Dunder Mifflin, los datos ya se encontraban en formato tabular y con valores atómicos, por lo que no fue necesario realizar grandes modificaciones para cumplir con 1FN. Sin embargo, se constató que existía redundancia en columnas como 'NombreDepartamento', 'PresupuestoDepartamento', 'NombreJefe', entre otras, que se repetían en múltiples filas debido a la relación de varios empleados con un mismo departamento.

### **Segunda Forma Normal (2FN)**

La Segunda Forma Normal (2FN) exige que todos los atributos que no sean clave dependan completamente de la clave primaria. En el conjunto de datos original, atributos como el 'NombreDepartamento', 'PresupuestoDepartamento' y 'NombreJefe' no dependían del identificador único del empleado, sino que pertenecían conceptualmente a otras entidades. Por esta razón, se procedió a descomponer la tabla original en tres nuevas tablas relacionadas:

1. Empleado (ID, Nombre, Apellido, Teléfono, Correo, IDDepartamento)

2. Departamento (IDDepartamento, NombreDepartamento, UbicaciónDepartamento, Presupuesto, IDJefe)

3. Jefe (IDJefe, NombreJefe, TeléfonoJefe)

Esta descomposición permite una mejor organización de los datos, elimina la redundancia, y facilita la integridad referencial en el sistema, ya que cada tabla representa una entidad independiente con relaciones claras entre sí.

## **Diagramas Lógico y Físico**

### **Diagrama Lógico**

El diagrama lógico representa las entidades y sus relaciones sin especificar detalles de implementación como tipos de datos. Se diseñaron las siguientes tres entidades: Empleado, Departamento y Jefe, con relaciones uno-a-muchos entre Departamento-Empleado y Jefe-Departamento.

EMPLEADO (IDEmpleado, NombreEmpleado, ApellidoEmpleado, Telefono, Correo, IDDepartamento)

DEPARTAMENTO (IDDepartamento, NombreDepartamento, UbicacionDepartamento, Presupuesto, IDJefe)

JEFE (IDJefe, NombreJefe, TelefonoJefe)

Diagrama

El contenido generado por IA puede ser incorrecto.

### **Diagrama Físico**

El diagrama físico incluye los tipos de datos, claves primarias y foráneas utilizadas en cada tabla, adaptadas para el motor de base de datos HeidiSQL:

-- Tabla Jefe

CREATE TABLE Jefe (

IDJefe INT AUTO\_INCREMENT PRIMARY KEY,

NombreJefe VARCHAR(100) NOT NULL,

TelefonoJefe VARCHAR(20)

);

-- Tabla Departamento

CREATE TABLE Departamento (

IDDepartamento INT AUTO\_INCREMENT PRIMARY KEY,

NombreDepartamento VARCHAR(100) NOT NULL,

UbicacionDepartamento VARCHAR(100),

Presupuesto DECIMAL(12,2),

IDJefe INT,

FOREIGN KEY (IDJefe) REFERENCES Jefe(IDJefe)

);

-- Tabla Empleado

CREATE TABLE Empleado (

IDEmpleado INT AUTO\_INCREMENT PRIMARY KEY,

NombreEmpleado VARCHAR(100) NOT NULL,

ApellidoEmpleado VARCHAR(100) NOT NULL,

Telefono VARCHAR(20),

Correo VARCHAR(100),

IDDepartamento INT,

FOREIGN KEY (IDDepartamento) REFERENCES Departamento(IDDepartamento)

);

**Inserción de Datos**

INSERT INTO Jefe (NombreJefe, TelefonoJefe)

VALUES ('Jan Levinson', '555-5678'),

('Michael Scott', '555-1234');

INSERT INTO Departamento (NombreDepartamento, UbicacionDepartamento, Presupuesto, IDJefe)

VALUES ('Ventas', 'Scranton', 150000.00, 1),

('Recepción', 'Scranton', 50000.00, 2);

INSERT INTO Empleado (NombreEmpleado, ApellidoEmpleado, Telefono, Correo, IDDepartamento)

VALUES

('Michael', 'Scott', '555-1234', 'michael.scott@dundermifflin.com', 1),

('Jim', 'Halpert', '555-2345', 'jim.halpert@dundermifflin.com', 1),

('Dwight', 'Schrute', '555-3456', 'dwight.schrute@dundermifflin.com', 1),

('Pam', 'Beesly', '555-4567', 'pam.beesly@dundermifflin.com', 2),

('Ryan', 'Howard', '555-5678', 'ryan.howard@dundermifflin.com', 1);

* **Tabla Departamento**

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

* **Tabla Empleado**

**Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.**

**Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.**

* **Tabla Jefe**

**Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.**

**Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.**

# **Sistema**

Correcto funcionamiento de Apache y MySQL

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

## **Instrucciones de uso**

### **Inicio del Sistema:**

El sistema comienza desde el archivo: index.php

Desde ese archivo se puede acceder a los formularios:

* Crear Jefe (crear\_jefes.php)
* Crear Departamento (crear\_departamento.php)
* Crear Empleado (crear\_empleado.php)

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

En la imagen del formulario de creación de jefe, se muestran dos campos de entrada:

**Nombre del Jefe**: se solicita el nombre de la persona que será registrada como jefe en la base de datos.

**Teléfono**: se ingresa el número de contacto del jefe.

Al presionar el botón "Crear", la información se envía mediante el método POST al archivo PHP correspondiente (crear\_jefes.php), el cual se conecta a la base de datos y almacena los datos en la tabla Jefe.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Este formulario permite ingresar la información de un nuevo departamento:

**Nombre**: nombre del departamento.

**Ubicación**: lugar donde se encuentra.

**Presupuesto**: monto asignado al departamento (validado para ser positivo).

**ID Jefe**: un menú desplegable dinámico muestra los jefes registrados para seleccionar uno como responsable del departamento.

Este formulario también se envía a través del método POST al archivo crear\_departamento.php, donde se registra la información en la tabla Departamento.

El campo desplegable se genera automáticamente consultando la tabla de jefes, mostrando los nombres para facilitar la selección.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Este formulario permite registrar nuevos empleados en el sistema asociándolos a un departamento existente.

**Campos del formulario:**

**Nombre**: nombre del empleado.

**Apellido**: apellido del empleado.

**Teléfono**: número de contacto.

**Correo**: correo electrónico del empleado.

**ID Departamento**: un menú desplegable que lista los departamentos registrados previamente.

Este menú desplegable se genera dinámicamente desde la base de datos, lo que garantiza que los empleados solo puedan ser asignados a departamentos válidos.

# **Conclusión**

La migración de los datos desde un archivo Excel a una base de datos relacional permitió organizar de forma eficiente la información de la sucursal de Dunder Mifflin. A través de la normalización, se eliminaron redundancias y se garantizó integridad referencial entre empleados, departamentos y jefes. Esta estructura será la base para un sistema ERP funcional, escalable y fácil de mantener. El uso de SQL Server permitió validar la consistencia de los datos y ejecutar consultas de prueba exitosamente.

# **Referencias bibliográficas**

Coronel, C., & Morris, S. (2017). \*Database systems: Design, implementation, & management\* (12th ed.). Cengage Learning.

Microsoft. (n.d.). \*SQL Server documentation\*. https://learn.microsoft.com/sql/sql-server/