



UNIVERSIDAD NACIONAL AUTÓNOMA DE  
MÉXICO  
FACULTAD DE INGENIERÍA  
INGENIERÍA EN COMPUTACIÓN



BASES DE DATOS  
GRUPO: 01

## PROYECTO FINAL

Alumnos:

- 
- López Aniceto Saúl Isaac
- López González Kevin
- 
- Ponce Soriano Armando

Profesor:

ING. Fernando Arreola Franco

24 de noviembre de 2021

# Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Plan de trabajo</b>	<b>2</b>
2.1. Descripción . . . . .	2
2.2. Plan de actividades . . . . .	2
2.3. Cronograma . . . . .	4
2.4. Aportaciones . . . . .	5
<b>3. Diseño</b>	<b>5</b>
3.1. Análisis de requerimientos . . . . .	5
3.2. Modelo conceptual . . . . .	5
3.2.1. Modelo Entidad-Relación . . . . .	6
3.3. Modelo lógico . . . . .	6
3.3.1. Representación Intermedia . . . . .	6
3.3.2. Transformación de MER a MR . . . . .	7
3.3.3. Modelo Relacional . . . . .	7
3.3.4. Normalización . . . . .	7
<b>4. Implementación</b>	<b>7</b>
4.1. Modelo físico . . . . .	7
4.1.1. IaaS . . . . .	7
4.2. Códigos . . . . .	7
4.3. DDL . . . . .	7
<b>5. Presentación</b>	<b>7</b>
5.1. Django . . . . .	7
5.1.1. Mapeo Relacional de Objetos . . . . .	7
5.1.2. Ejecutar SQL personalizado directamente . . . . .	8
5.2. Diseño . . . . .	8
<b>6. Conclusiones</b>	<b>8</b>

## 1. Introducción

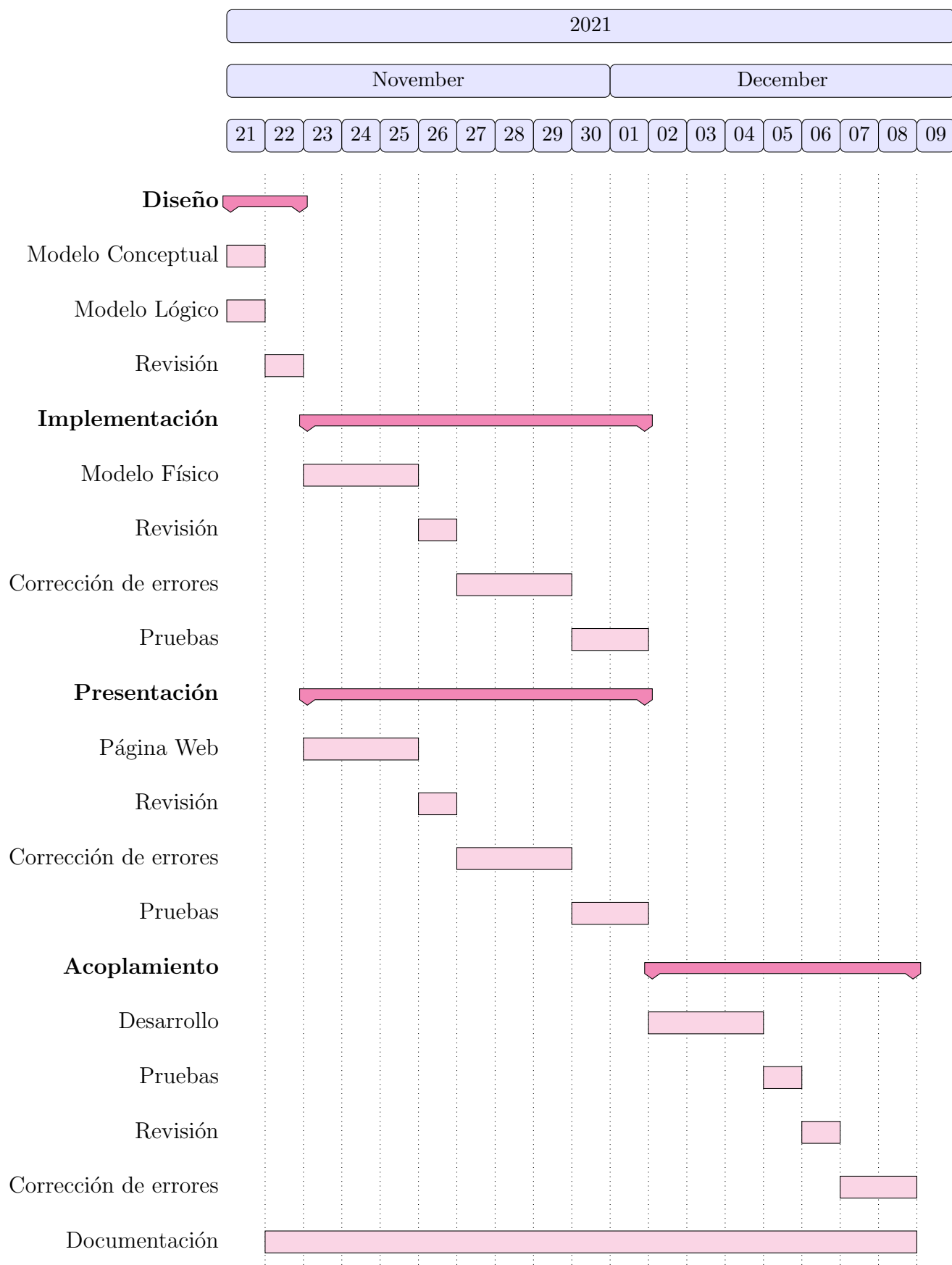
## 2. Plan de trabajo

### 2.1. Descripción

### 2.2. Plan de actividades



### 2.3. Cronograma



## 2.4. Aportaciones

	Diseño	Implementación	Presentación	Acoplamiento	Documentación
Kevin López	✓		✓	✓	✓

## 3. Diseño

### 3.1. Análisis de requerimientos

### 3.2. Modelo conceptual

#### Entidades

- PROVEEDOR: { id\_Proveedor, razón social, domicilio (estado, código postal, colonia, calle y número), nombre, teléfonos }
- CLIENTE: { RFC, nomre (nombre, ap\_Paterno, ap\_Materno), domicilio (estado, código postal, colonia, calle y número), emails }
- INVENTARIO: { id\_Inventario, precio\_compra, fecha\_compra, cantidad\_ejemplares }
- PRODUCTO: { código\_Barras, marca, descripción, precio, categoria }
- VENTA : { num\_venta, fecha\_venta, pago\_Total, cantidad\_articulo, pago\_total\_Articulo }

#### Relaciones

- Un proveedor surte a muchos inventarios.
- Un inventario es surtido por muchos proveedores.
- Un inventario almacena muchos productos.
- Un producto es almacenado por un inventario.
- Una venta contiene muchos productos.
- Un producto es contenido es muchas ventas.
- Un cliente concreta muchas ventas.
- Una venta es concretada por un cliente.

### 3.2.1. Modelo Entidad-Relación

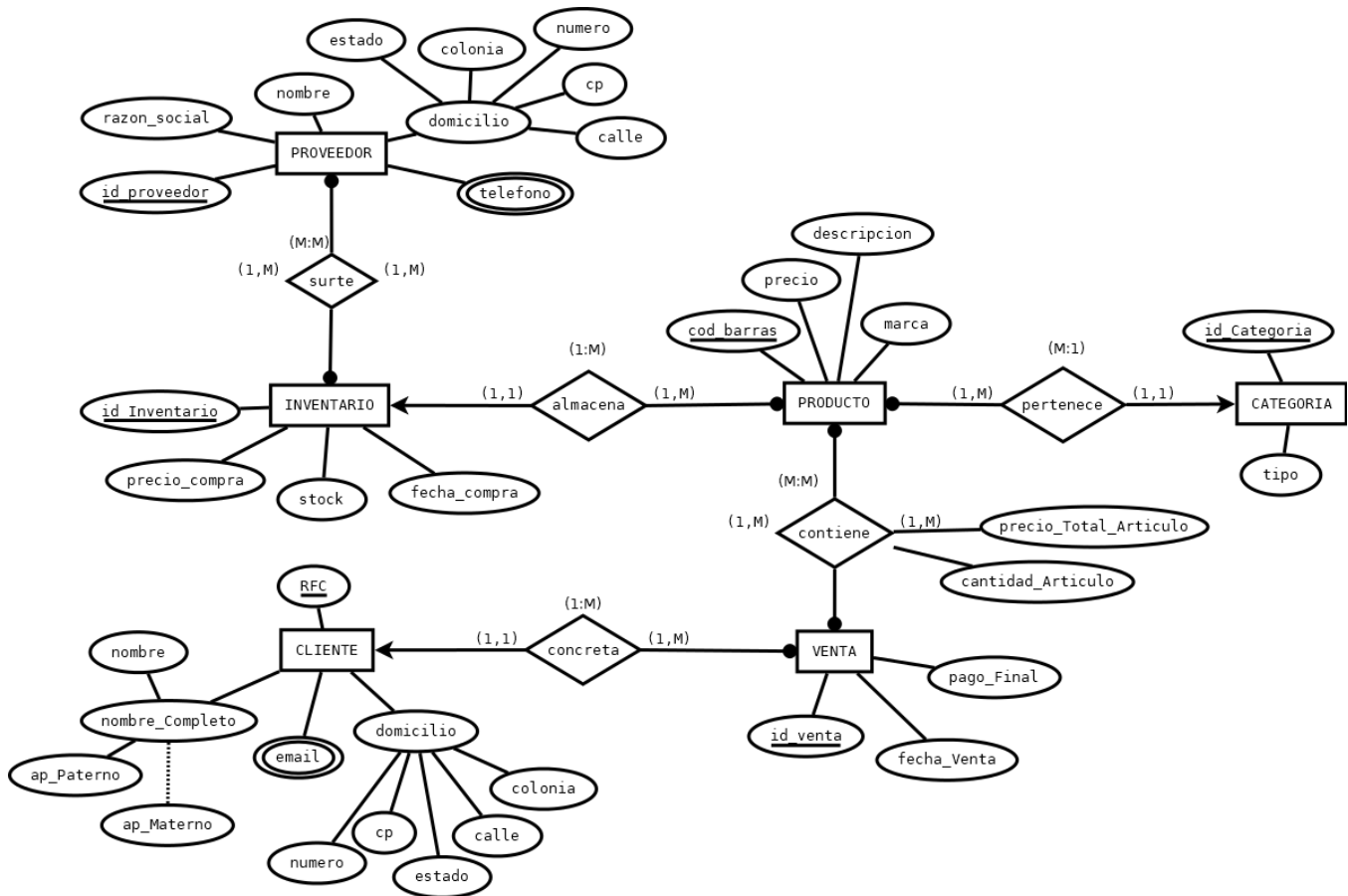


Figura 1: Modelo Entidad-Relación.

## 3.3. Modelo lógico

### 3.3.1. Representación Intermedia

- **PROVEEDOR**: { id\_proveedor smallint (PK), nombre varchar 50, razón social varchar 50, estado varchar 50, colonia varchar 50, numero smallint, cp smallint, calle varchar 50 }
- **TELEFONO**: { teléfono bigint(PK), id\_proveedor smallint (FK) }
- **INVENTARIO**: { id\_inventario smallint (PK), precio\_compra decimal (10,2), stock smallint, fecha\_compra date }
- **SURTE**: { [id\_Proveedor smallint (FK), id\_inventario smallint (FK)] (PK) }
- **PRODUCTO**: { cod\_barras integer PK, id\_categoria smallint FK, precio smallint NOT NULL, marca varchar(20) NOT NULL, descripcion varchar(50), id\_inventario smallint (FK) }
- **CATEGORÍA**: { id\_categoria smallint PK, tipo varchar(20) NOT NULL }

- CLIENTE: {RFC varchar(13) (PK), nombre varchar(20), ap\_paterno varchar (20), ap\_materno varchar (20) (N), cp smallint, numero smallint, estado varchar (32), calle varchar (32), colonia varchar (32)}
- EMAIL: {RFC varchar(13) (FK), email varchar (64) (PK)}
- VENTA: {id\_venta int(PK), fecha\_venta date, pago\_final decimal(7,2), RFC varchar(13)(FK)}
- CONTIENE: { [cod\_barras int , id\_venta int](PK)(FK), precioTotalArt decimal(7,2), cantidad articulo int }

### 3.3.2. Transformación de MER a MR

### 3.3.3. Modelo Relacional

### 3.3.4. Normalización

## 4. Implementación

### 4.1. Modelo físico

#### 4.1.1. IaaS

### 4.2. Códigos

### 4.3. DDL

## 5. Presentación

Una interfaz gráfica es un programa que nos permite manipular información a través de objetos gráficos que proporcionen un entorno visual, con el fin facilitar la interacción del usuario con la computadora.

En este caso, se optó por desarrollar una página web como interfaz gráfica que permita la manipulación de información de nuestra base de datos.

### 5.1. Django

Django es un framework de Python de alto nivel que permite diseñar aplicaciones web de una forma rápida, limpia y pragmática. Además, ayuda a los desarrolladores a evitar muchos errores de seguridad comunes, como la inyección de SQL, las secuencias de comandos entre sitios, la falsificación de solicitudes entre sitios y el secuestro de clics.

#### 5.1.1. Mapeo Relacional de Objetos

El Mapeo Relacional de Objetos o ORM (*Object Relational Mapping*), es una tecnología que soluciona el desajuste entre las bases de datos relacionales y orientadas a objetos.

Por lo general, asigna una clase a una tabla uno a uno. Cada instancia de la clase corresponde a un registro en la tabla y cada atributo de la clase corresponde a cada campo en la tabla. ORM



proporciona una asignación a la base de datos, en lugar de escribir código SQL directamente, solo es necesario manipular los datos de la base de datos como un objeto operativo.

Si bien, el ORM es una herramienta muy útil, no se utilizará en este proyecto, ya que preferimos escribir directamente la sentencia SQL para comunicarnos con la base de datos.

### 5.1.2. Ejecutar SQL personalizado directamente

El objeto **django.db.connection** representa la conexión por defecto entre django y la base de datos. Las funciones que utilizamos para la comunicación con la base de datos son las siguientes:

- **connection.cursor()**

Para obtener un objeto cursor.

- **cursor.execute(sql, [params])**

Para ejecutar sentencias SQL.

- **cursor.fetchall()**

Para devolver las filas resultantes de la consulta.

## 5.2. Diseño

## 6. Conclusiones

- 

- López González Kevin

Bla bla bla

-