

---

## PROYECTO NO. 3

---

201807154 – Denny Alexander Chalí Miza

### Resumen

Para la realización del proyecto No. 3 se realizó una aplicación web cuyas funciones son las de analizar dos archivos de texto plano con formato XML, los cuales cargan al sistema los servicios de infraestructura de los servicios que presta una empresa, así como los consumos que realizan cada uno de sus clientes. La función principal del programa es la de realizar facturas en base a los servicios prestados a los clientes. Otra función de la aplicación es la de crear nuevos datos y guardar la información en la base de datos.

Para la solución del problema se hizo uso de dos Framework web. Para el lado del servidor (Backend) se utiliza Flask, en la cual se realizan todas las funciones lógicas del sistema como: análisis de datos, recepción y envío de datos y la escritura de la información en las bases de datos. Lo relacionado con la parte gráfica se hizo uso del framework Django, en el cual se envía la información al servidor para su procesamiento y se encarga de mostrar los datos cargados al sistema.

### Palabras clave

API, Controlador, Petición, Servidor

### Abstract

*For the realization of project No. 3, a web application was developed whose functions are to analyze two plain text files in XML format, which upload to the system the infrastructure services of the services provided by a company, as well as the consumption made by each of its customers. The main function of the program is to create invoices based on the services provided to customers. Another function of the application is to create new data and save the information in the database.*

*Two web frameworks were used to solve the problem. For the server side (Backend) Flask is used, in which all the logical functions of the system are performed such as: data analysis, receiving and sending data and writing the information in the databases. The graphical part is related to the Django framework, which sends the information to the server for processing and is responsible for displaying the data uploaded to the system.*

### Keywords

API, Controller, Request, Server.

## Introducción

Se desarrollo una aplicación que se encarga de facturar detalladamente los servicios de infraestructura que aprovisiona la empresa de Tecnologías Chapinas, S.A a sus clientes.

La estructura de la tecnología consiste en crear configuraciones de infraestructuras que agrupan recursos necesarios para que una empresa pueda construir las arquitecturas de despliegue de aplicaciones que requiera.

Para ello se creó un software web compuesto por la parte grafica (Frontend) y la parte encargada de almacenar y procesar los datos (Backend). Para la parte gráfica se utilizó el Framework web Django, la cual se encarga de enviar los datos al Backend, así como de visualizar y añadir nuevos datos al sistema. La parte lógica del sistema es realiza por el Backend, el cual se creo usando el Framework web Flask, la cual se encarga de procesar y almacenar la información en las bases de datos, así como de enviar los datos cuando se le solicite.

## Desarrollo del tema

### a. Preparando el entorno de trabajo

Para la resolución del problema se utilizó el lenguaje de programación Python, el cual deberá de descargar e instalar en su ordenador, para el correcto funcionamiento del programa se recomienda instalar una versión superior a la 3.6.

Flask. Es un Framework escrito en Python y concebido para facilitar el desarrollo de Aplicaciones Web bajo el patrón MVC

Para instalar Flask vamos a utilizar **pip**. Así que simplemente deberemos de escribir en nuestra línea de comandos lo siguiente:

```
$ pip install Flask
```

Puede ser que para la instalación necesites ser administrador. En ese caso ejecuta:

```
$ sudo pip install Flask
```

**Django.** Django es un framework web de alto nivel que permite el desarrollo rápido de sitios web seguros y mantenibles. Se encarga de gran parte de las complicaciones del desarrollo web.

Para instalar Django vamos a utilizar **pip**. Así que simplemente deberemos de escribir en nuestra línea de comandos lo siguiente:

```
$ pip install Django
```

**Requests.** Es una biblioteca de solicitudes para realizar solicitudes HTTP en Python

Para instalar Requests vamos a utilizar **pip**. Así que simplemente deberemos de escribir en nuestra línea de comandos lo siguiente:

```
$ pip install requests
```

**Flask-Cors.** El Intercambio de Recursos de Origen Cruzado es un mecanismo que permite configurar un servidor para obtener permisos para acceder a recursos seleccionados de dicho servidor a un origen distinto (dominio).

Para instalar Requests vamos a utilizar **pip**.

```
$ pip install flask-cors
```

## b. Ejecución del programa

**Posicionarse en la carpeta (CMD).** Una vez el entorno de trabajo está listo podemos iniciar el programa, para ello abrimos una pestaña en el símbolo del Sistema (CMD) y navegamos hasta estar en la carpeta donde se ubica los ejecutables del programa:

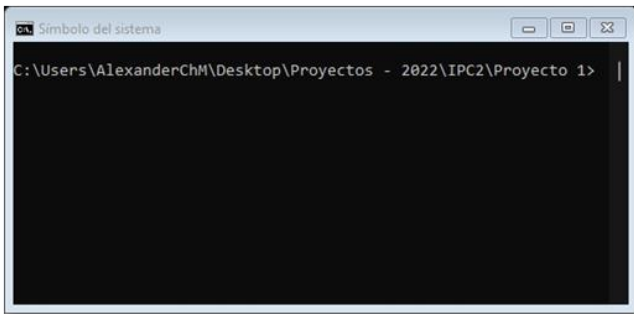


Figura 1. Ejemplo de ubicación en carpeta

Fuente: elaboración propia.

**Frontend.** Para ejecutar el servidor web del Frontend nos ubicamos en la carpeta del mismo nombre e ingresamos el siguiente comando.

```
$ python manage.py runserver
```

**Backend.** Para ejecutar el servidor web del Backend nos ubicamos en la carpeta del mismo nombre e ingresamos el siguiente comando.

```
$ python main.py
```

Es necesario iniciar ambos servidores para poder acceder a todas las funciones del sistema porque son dependiente entre sí.

## c. Funciones del programa

**Pantalla de inicio.** Es la pantalla que se muestra al ingresar a la dirección interna que nos proporciona

el servidor Frontend en la cual as muestra la siguiente ventana:

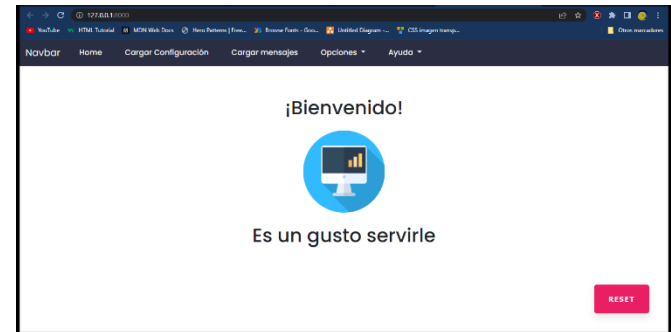


Figura 2. Pantalla de inicio.

Fuente: elaboración propia.

**Home.** Nos dirige a la pantalla de inicio.

**Cargar Configuración.** Nos dirige a una ventan para gestionar el envío de los mensajes de entrada con extensión .xml con una o varias solicitudes de creación del mensaje.

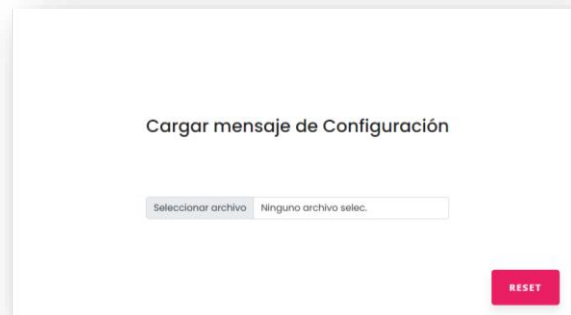


Figura 3. Carga de archivos.

Fuente: elaboración propia.

Si la carga de archivos se realizó con éxito se desplegará en la pantalla un mensaje mostrando la cantidad de cliente e instancias agregadas.

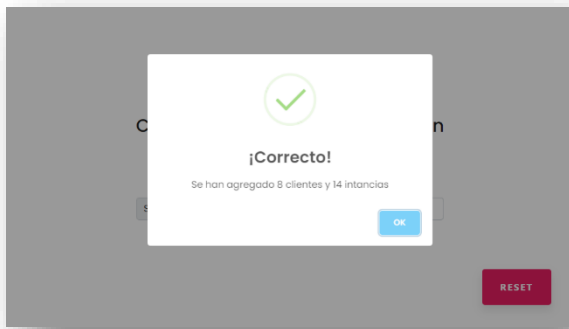


Figura 4. Mensaje de la solicitud realizada

Fuente: elaboración propia.

**Cargar mensajes de consumo.** Se desplegará una pantalla para gestionar el envío del mensaje de entrada con extensión .xml con una o varias solicitudes de consumo de recursos del sistema.

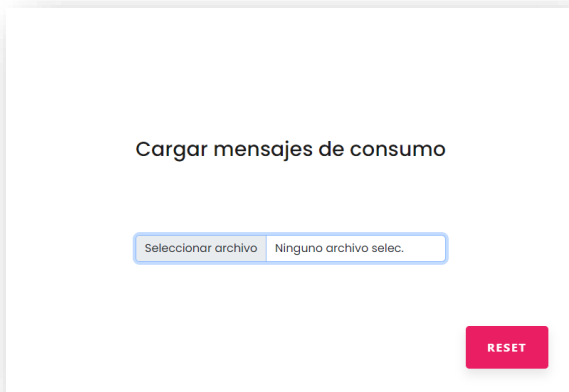


Figura 5. Carga de archivos.

Fuente: elaboración propia.

Si la carga de archivos se realizó con éxito se desplegará en la pantalla un mensaje mostrando la cantidad de consumos agregados al sistema.

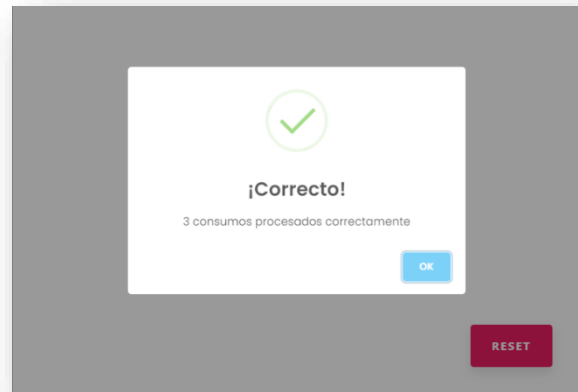


Figura 6. Mensaje de la solicitud realizada

Fuente: elaboración propia.

**Opciones.** En este apartado se debe de tener las siguientes:

**Consultar Datos:** Al seleccionar esta opción se podrá visualizar la estructura de información que actualmente maneja el sistema, es decir, podrá ver las categorías, recursos y configuraciones disponibles, o bien, los clientes e instancias registradas, así como los recursos pendientes de facturar por cada Instancia.

Categorías			
Recursos			
Clientes			
ID: 1			
Nombre: Maquinas virtuales			
Descripción: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua			
Carga: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua			
Configuraciones			
ID	Nombre	Descripción	Recursos
1	Basico I	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua	ID Cantidad
			1 2
			2 2
2	Basico II	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua	ID Cantidad
			1 2
			2 2
			3 1

Figura 7. Categorías existentes en el sistema.

Fuente: elaboración propia

Categorías Recursos Clientes					
ID	Nombre	Abreviatura	Métrica	Tipo	Costo hora
1	Procesador	CPU	unidades	Hardware	2
2	Memoria de acceso aleatorio	Ram	Gigabytes	Hardware	4
3	Windows 10 pro	win 10	Gigabytes	Software	0.50
4	Tarjeta gráfica	GPU	Gigabytes	hardware	2.50
5	Dispositivo de almacenamiento	SSD	Gigabytes	hardware	1.50

Figura 8. Recursos disponibles.

Fuente: elaboración propia

**Creación de nuevos datos:** Al seleccionar esta opción se podrá crear nueva información, es decir, podrá crear nuevas categorías, nuevos recursos, nuevas configuraciones, nuevos clientes o registrar nuevas instancias, cancelar instancias.

## Datos del Desarrollador



**Nombre:** Denny Alexander Chalí Miza  
**Carnet:** 201807154  
**Correo:** 2838484560404@ingenieria.usac.edu.gt

Figura 10. Datos del desarrollador.

Fuente: elaboración propia.

### Crear y agregar nuevos datos

Recursos

Categorías

Clientes

Configuración

Instancias

ID

Nombre

Abreviatura

Métrica

Tipo

Valor por hora

Crear

Figura 9. Creación de nuevos recursos.

Fuente: elaboración propia

**Nota:** Para la creación de los nuevos datos se deberá de seleccionar y llenar cada uno de los campos del formulario.

**Proceso de facturación:** Al seleccionar esta opción se podrá elegir un rango de fechas y se generará una factura para cada cliente por todos los recursos consumidos no facturados previamente.

**Ayuda.** Desplegará 2 opciones, una para visualizar información del desarrollador y otra para visualizar la documentación del programa.

## Conclusiones

- Los Frameworks Web son herramientas poderosas que nos permiten crear proyectos simples o complejos por su versatilidad.
- Existen muchos framework orientados a distintos lenguajes, pero los orientados al Lenguaje Python son completos y fáciles de usar gracias a toda la documentación disponible.
- La utilización de un framework en el desarrollo de una aplicación implica un cierto tiempo inicial de aprendizaje, aunque a largo plazo es probable que facilite tanto el desarrollo como el mantenimiento.

## Referencias bibliográficas

Lewis, John; Loftus, William (2009). *Java Software Solutions Foundations of Programming Design* 6th ed. Pearson Education Inc.

Chacón Sartori, Camilo. *Computación y programación funcional: introducción al cálculo lambda y la programación funcional usando Racket y Python*. [Barcelona]: Marcombo.

<http://tutorialms-dos.bligoo.com>,  
"Listas Enlazadas Simples, Circulares... Y Árboles Binarios", 2002.

## Anexos

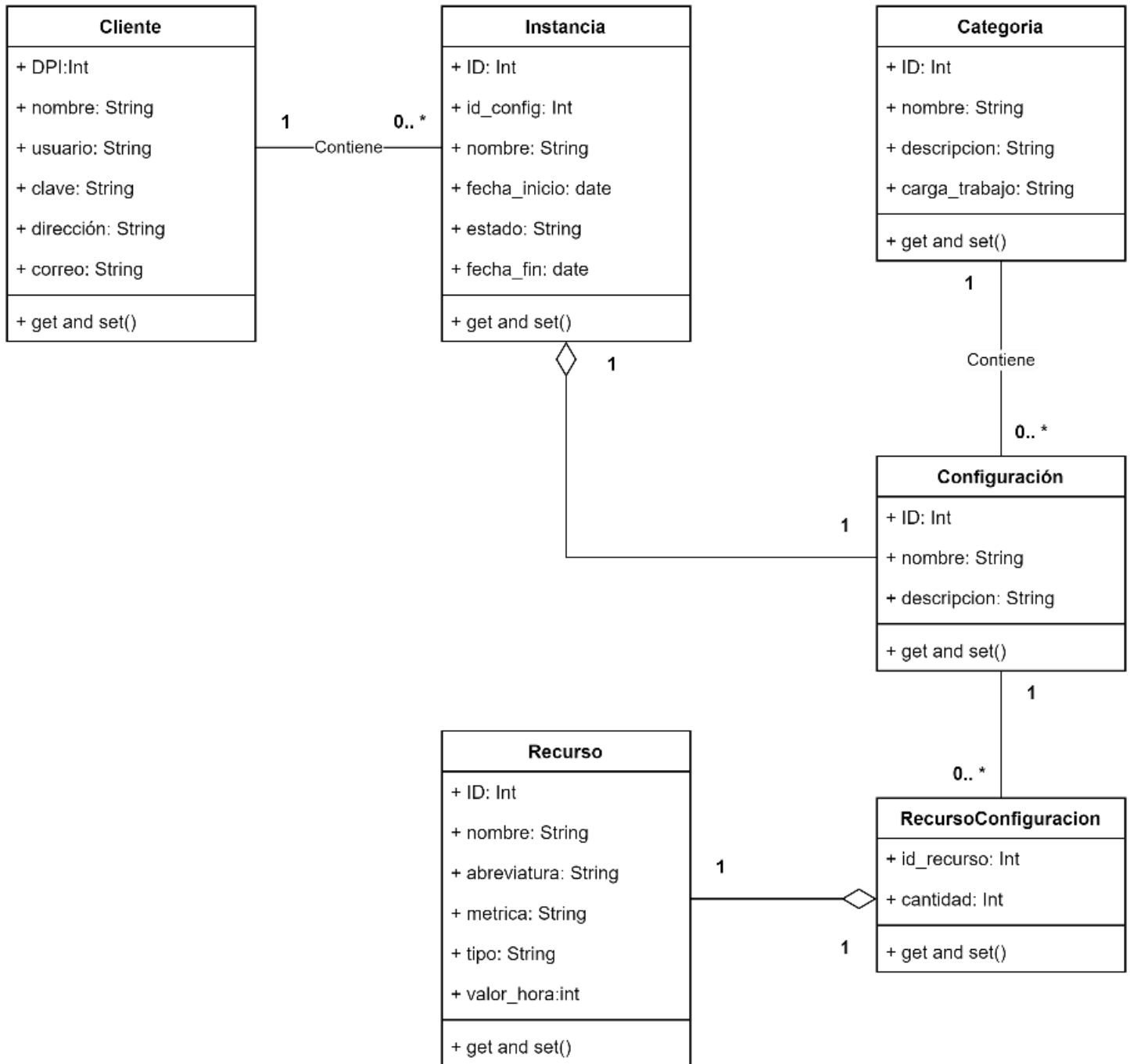


Figura 11. Diagrama de clases para la solución del problema

Fuente: elaboración propia

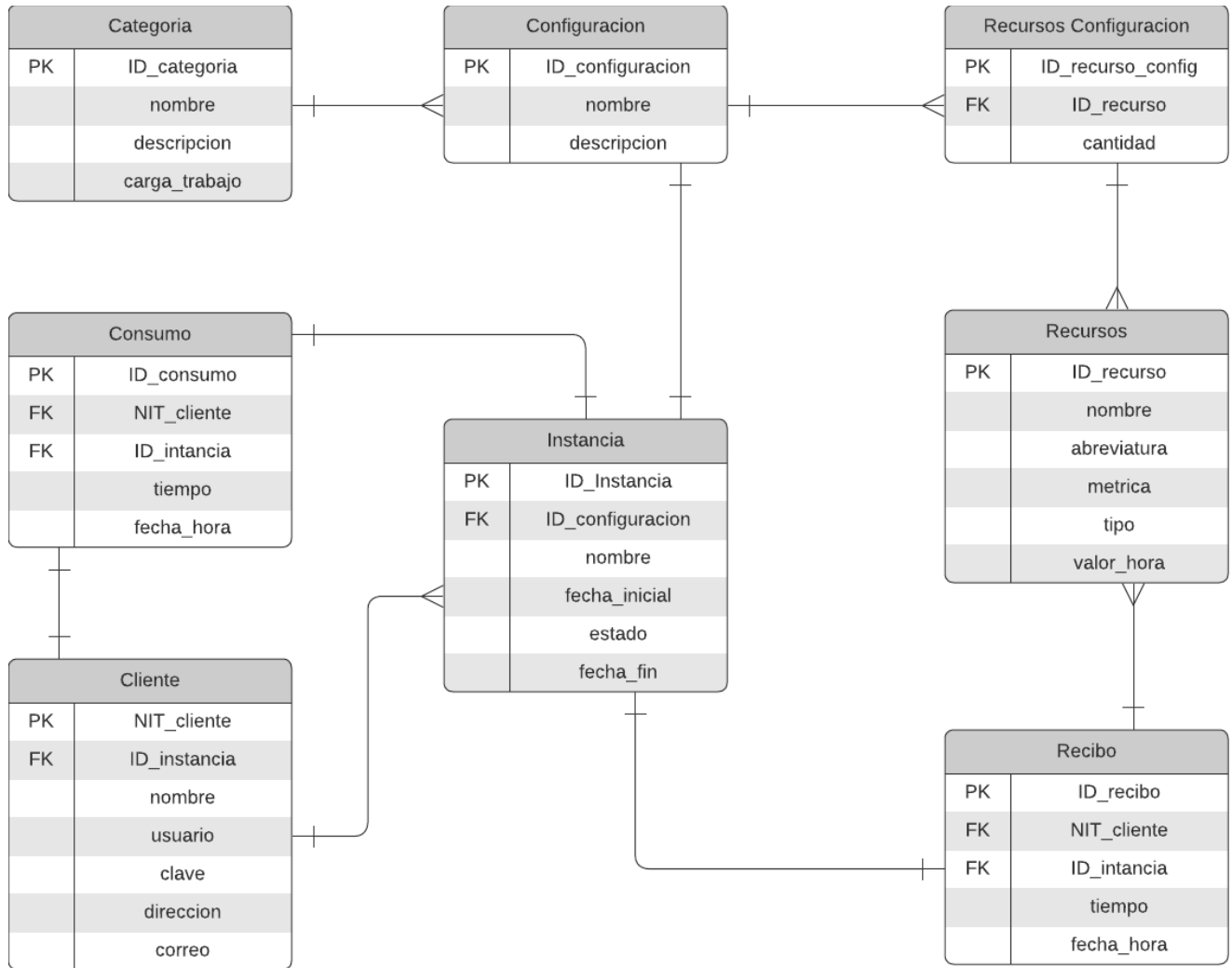


Figura 12. Diagrama del modelo relacional utilizado para la solución del problema

Fuente: elaboración propia