



Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computación  
IC-4302 - Bases de Datos II  
Grupo 20

Proyecto 1  
Documentación Proyecto

Estudiantes:  
José Manuel Quesada Carvajal - 2021082885  
Emmanuel Lopez Ramirez - 2018077125

Profesora: María Auxiliadora Mora

Fecha de entrega: 23 de setiembre del 2022

## Introducción

Actualmente las bases de datos son de mucha utilidad en las empresas y en los servicios que muchas personas utilizan en sus vidas diarias. En muchos casos la robustez y confiabilidad de estas son tema serio en las compañías ya que de estas dependen muchos dólares.

En función de este proyecto, se tiene el problema de un restaurante el cual quiere mantener registro de sus productos y facturas en una base de datos confiable donde se puedan realizar consultas y guardar datos por medio de una interfaz fácil de usar. Ante este problema, y ante la gran importancia de las bases de datos en la cotidianidad y en la informática, se tiene la intención de solventar este problema realizando una aplicación web que se conecte a dos bases de datos, una creada para almacenar los datos del restaurante, cliente y ventas, y otra que almacena metadatos y punteros a fotografías de los productos.

## Descripción del sistema

Cómo fue mencionado en la introducción, se realizará una aplicación web que se conecte a dos bases de datos, para solucionar el problema de un restaurante que requiere de un sistema para facturar y mostrar platillos a clientes. A continuación se detalla de manera más técnica el sistema:

En primer lugar se tendrá una base de datos relacional en **Oracle**, corriendo en Oracle Linux. Esta base de datos almacenará información de clientes, productos y facturas.

Por otro lado, para tener control de productos y mostrarlos a los usuarios, también se contará con una base de datos no relacional en **MongoDB** que almacenará punteros y metadatos de imágenes de los productos que vende el restaurante.

Por último una interfaz web permitirá al restaurante acceder a la información de ambas bases para consultas, así cómo también permitirá almacenar datos en estas de una forma fácil de usar e intuitiva para el usuario.

## Objetivos y alcances

- Diseñar e implementar una aplicación web que solviente requerimientos de un sistema de información de la vida real utilizando Java, Apache Tomcat, Oracle y MongoDB.
  - Utilizar y evaluar las tendencias actuales en el campo de la tecnología de base de datos.
  - Comprender el funcionamiento de una base de datos no relacional.
  - Emplear el conocimiento aprendido para realizar una base de datos relacional en Oracle.
  - Reforzar la programación en Java, y en HTML/CSS.

Por la complejidad del proyecto considerando el tiempo, se limitará el proyecto a tener cuatro clases de la vida real como máximo en la base de datos no relacional, las cuales van a estar relacionadas con el producto que vende el restaurante, el cliente y la venta.

## Conceptualización del sistema

### Casos de uso a Implementar

Se tienen los siguientes casos de uso que se implementarán en la aplicación:

Nombre del Caso de Uso	Facturar compra
Actores	Empleado
Precondiciones	Debe tener un usuario y contraseña para ingresar
Post-condiciones	El empleado realiza la factura de compra y el sistema la registra en la base de datos
Flujo Básico	<ol style="list-style-type: none"><li>1. El sistema muestra al empleado campos para ingresar su usuario y contraseña, se tiene un botón de ingresar.</li><li>2. El empleado ingresa sus datos válidos y toca el botón de ingresar.</li><li>3. El sistema muestra campos para seleccionar productos desde un dropdown menu donde se</li></ol>

	<p>elige la cantidad. Se tiene un campo para ingresar la cédula de cliente ya registrado. Se tiene un botón de facturar.</p> <ol style="list-style-type: none"> <li>4. El empleado ingresa los datos y presiona el botón de facturar.</li> <li>5. Se guarda la información en la base de datos relacional. Se le muestra al empleado la factura.</li> </ol>
Flujo Alterno	<p>Si el empleado no ingresa un usuario y contraseña válidos.</p> <ol style="list-style-type: none"> <li>1. El sistema informa al empleado que se ingresó información inválida.</li> <li>2. El sistema le pide que vuelva a ingresar la información correcta y vuelve a pedir usuario y contraseña.</li> </ol> <p>Si el empleado ingresa una cédula de cliente inválida o no existente</p> <ol style="list-style-type: none"> <li>1. Se le muestra al empleado que ese cliente no existe.</li> <li>2. Se le muestra un botón que en caso de ser presionado lo redirige a registrar el usuario.</li> </ol>
Resultado	El empleado registra la venta, el sistema la guarda en la base de datos. El sistema muestra al vendedor la factura creada.

Nombre del Caso de Uso	Registrar Cliente
Actores	Empleado y Cliente
Precondiciones	La cédula del cliente no puede estar registrada en el sistema previamente
Post-condiciones	Se crea un cliente nuevo en el sistema, se le muestra un mensaje de confirmación al usuario
Flujo Básico	<ol style="list-style-type: none"> <li>1. El sistema muestra campos para los detalles del cliente a crear. Se tiene el campo de cédula, nombre, apellido 1, apellido 2, correo y teléfono. Se tiene un botón de crear.</li> <li>2. El usuario ingresa los datos y presiona el botón</li> </ol>

	<p>de registrar.</p> <p>3. Se guarda la información en la base de datos relacional. Se le muestra al usuario una confirmación de que el usuario fue creado.</p>
Flujo Alterno	<p>Si la cédula del empleado ya está en el sistema</p> <p>1. Se le muestra al usuario que el cliente ya está registrado, se le pide que por favor ingrese una cédula no registrada.</p> <p>Si la cédula no es válida</p> <p>1. Se le informa al usuario que la cédula no sigue la forma '1-1111-1111' y se le pide de nuevo la información.</p>
Resultado	<p>Se le muestra al usuario un mensaje confirmando que la cuenta fue creada. El sistema guarda al cliente en la base de datos relacional.</p>

Nombre del Caso de Uso	Eliminar Cliente
Actores	Empleado
Precondiciones	Debe estar la cédula del cliente registrada en el sistema
Post-condiciones	Se borra registro del cliente en el sistema, se le muestra un mensaje de confirmación al usuario
Flujo Básico	<p>1. El sistema muestra al empleado campos para ingresar su usuario y contraseña, se tiene un botón de ingresar.</p> <p>2. El empleado ingresa sus datos válidos y toca el botón de ingresar.</p> <p>3. El sistema muestra el campo de cédula y un botón de eliminar</p> <p>4. El usuario ingresa los datos y presiona el botón de eliminar.</p> <p>5. Se elimina el registro del cliente de la base de datos relacional.</p>
Flujo Alterno	<p>Si la cédula del empleado no está en el sistema</p> <p>1. Se le muestra al usuario que el cliente no está registrado, se le pide que por favor ingrese una cédula registrada.</p> <p>Si la cédula no es válida</p>

	2. Se le informa al usuario que la cédula no sigue la forma '1-1111-1111' y se le pide de nuevo la información.
Resultado	Se le muestra al usuario un mensaje confirmando que el cliente fue borrado. El sistema elimina al cliente en la base de datos relacional.

Nombre del Caso de Uso	Consultar productos
Actores	Cliente
Precondiciones	Ninguna.
Post-condiciones	Se le muestra al cliente los productos junto a una foto de este
Flujo Básico	<ol style="list-style-type: none"> <li>1. El sistema muestra un botón de Ver Comida.</li> <li>2. El usuario da click a este, se le muestra toda la lista de productos cada uno con el nombre, precio y una imagen de este.</li> </ol>
Flujo Alternativo	<p>En caso de que no se tenga productos</p> <ol style="list-style-type: none"> <li>1. Se le muestra al usuario un mensaje indicando que no se pueden ver los productos en este momento.</li> </ol>
Resultado	El sistema enseña al usuario los productos que ofrece el restaurante con sus detalles y una foto obtenida por medio de punteros en la base de datos no relacional

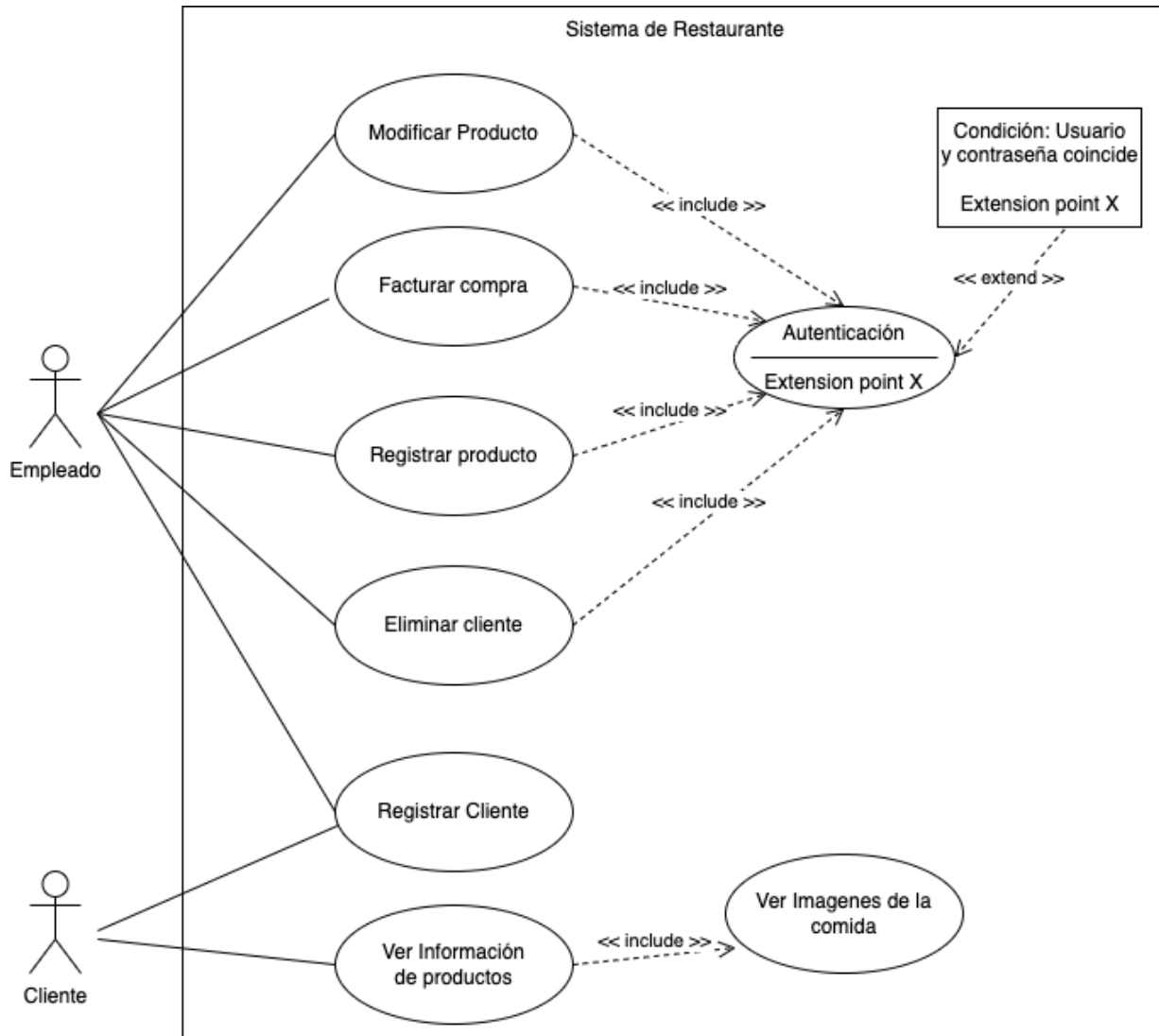
Nombre del Caso de Uso	Registrar producto
Actores	Empleado
Precondiciones	El mismo producto no puede estar registrado ya en el sistema
Post-condiciones	Se le muestra al empleado el producto agregado.

Flujo Básico	<ol style="list-style-type: none"> <li>1. Se le muestra al usuario los campos de nombre de producto y precio. Se tiene un botón de registrar</li> <li>2. El usuario ingresa los datos de los campos y presiona el botón de registrar.</li> <li>3. Se guarda el producto nuevo.</li> <li>4. Se le pide al usuario una imagen del producto y se carga.</li> <li>5. Se le confirma al usuario que se registró la imagen.</li> </ol>
Flujo Alternativo	<p>En caso de que no se tenga productos</p> <ol style="list-style-type: none"> <li>2. Se le muestra al usuario un mensaje indicando que no se pueden ver los productos en este momento.</li> </ol>
Resultado	El sistema enseña al usuario el producto que fue añadido con sus detalles y una foto obtenida por medio de punteros en la base de datos no relacional

Nombre del Caso de Uso	Modificar producto
Actores	Empleado
Precondiciones	El mismo producto tiene que estar registrado ya en el sistema
Post-condiciones	Se le muestra al empleado el producto modificado.
Flujo Básico	<ol style="list-style-type: none"> <li>1. El sistema muestra al empleado campos para ingresar su usuario y contraseña, se tiene un botón de ingresar.</li> <li>2. El empleado ingresa sus datos válidos y toca el botón de ingresar.</li> <li>3. Se le muestra al usuario el campo de nombre de producto. Se tiene un botón de modificar.</li> <li>4. El usuario presiona el botón de modificar y se le muestran los campos de nombre de producto y precio. Se tiene un botón de registrar</li> <li>6. El usuario ingresa los datos de los campos y presiona el botón de registrar.</li> <li>7. Se guarda el producto nuevo.</li> <li>8. Se le pide al usuario una imagen del producto y</li> </ol>

	<p>se carga.</p> <p>9. Se le confirma al usuario que se modificó el producto.</p>
Flujo Alterno	<p>En caso de que no se tenga ese productos</p> <ol style="list-style-type: none"> <li>1. Se le muestra al usuario un mensaje indicando que no existe dicho producto.</li> <li>2. Se le pide de nuevo ingresar un producto válido.</li> </ol>
Resultado	<p>El sistema enseña al usuario el producto que fue modificado con sus detalles y una foto obtenida por medio de punteros en la base de datos no relacional</p>





## Diseño del Sistema

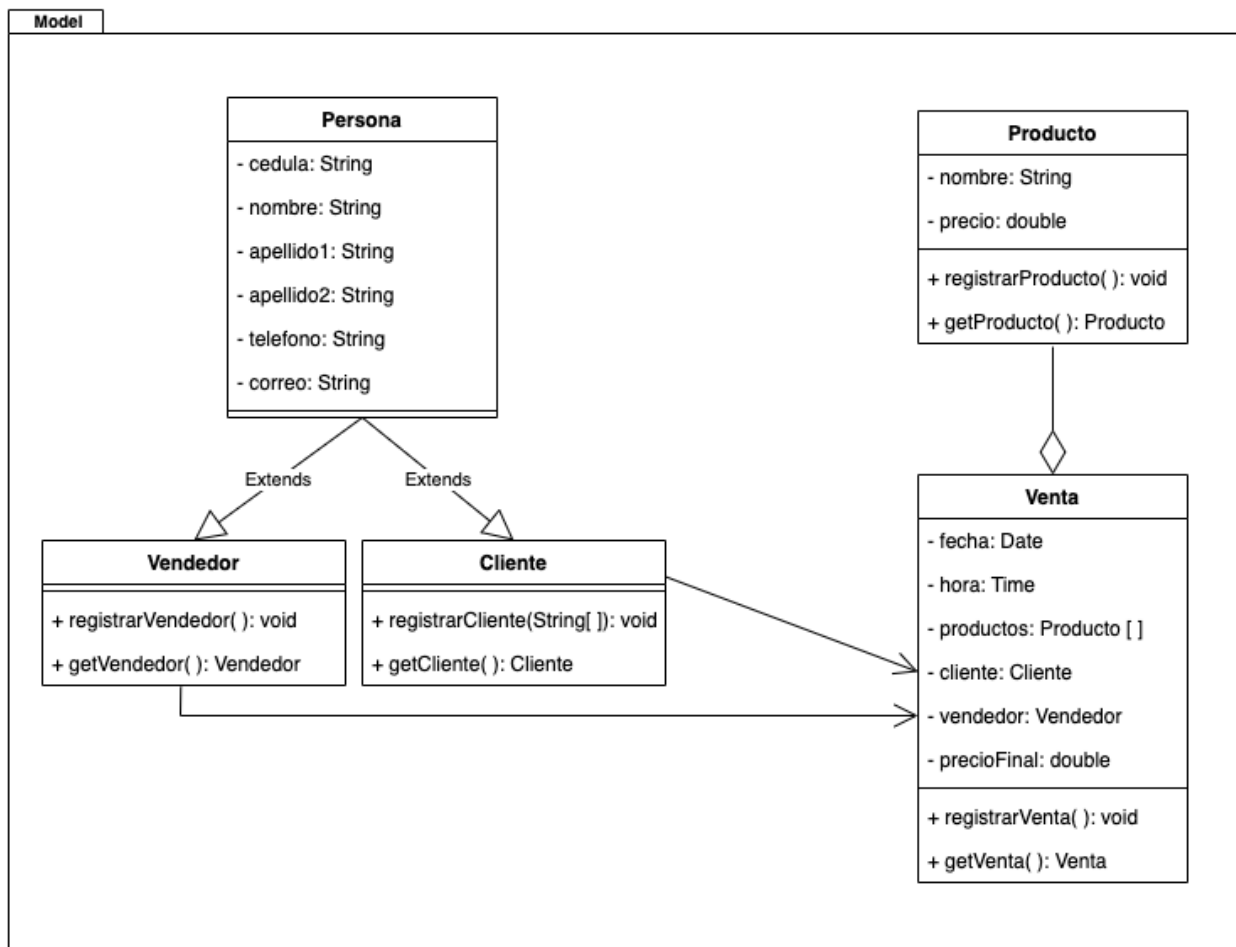
Este diseño consiste en determinar cómo funcionará la base de datos relacional en Oracle Database. La tabla personas es una tabla general donde se puede dividir en dos tipos normalizados para que pueda ser un vendedor(Empleado del restaurante) y un cliente de manera simultánea si fuese el caso, la otra tabla es de correos, ya que para evitar la atomicidad de los datos como lo es el correo no esté limitado a un solo correo por persona, Por otra parte la tabla venta registra todo lo relacionado con la facturación de los productos con su respectivo vendedor para su comisión y el cliente pueda hacer reclamos a su respectivo vendedor por medio de la factura a modo hipotético.

Este modelo posee una interfaz web en CCS/HTML/JS utilizando el framework React para desarrollar el frontend, y se utiliza Java como lenguaje controlador y un modelado

basado en el diagrama de modelo de clases, esto propone una arquitectura de modelo-vista-controlador(MVC) con ayuda de SpringBoot para un organizado eficiente entre las tareas de cada parte de la aplicación de este proyecto. El frontend y backend se comunican por medio de REST APIs que envían los request de las operaciones CRUD hacia la base de datos.

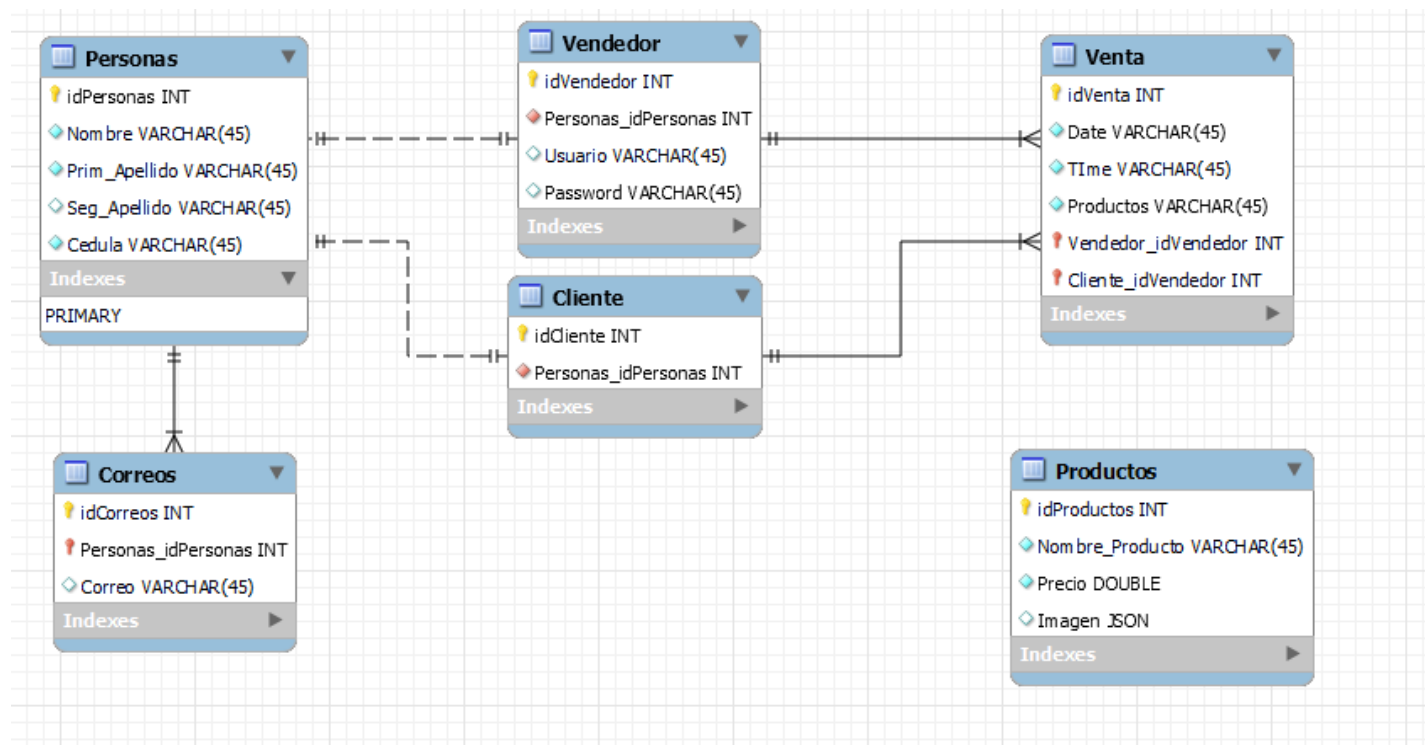
## Modelo de clases

Se tiene el siguiente modelo de clases en UML:



Como se ve en este modelo, se tienen cuatro clases de la vida real, **Vendedor**, **Cliente**, **Venta** y **Producto**.

## Diagrama Entidad - Relación



## Entidades en base relacional y no relacional

Las entidades a utilizar en Oracle database son las siguientes: personas, vendedor cliente, venta y correos.

La tabla Productos se almacena en la base de datos no relacional

Este último contempla dentro de la base de datos un dato de tipo imagen que representará el platillo que vende el restaurante y por lo tanto para una base de datos no relacional es un dato que solo será procesado por el cliente para el correcto flujo de la página web.

## Diccionario de datos

El siguiente diccionario de datos contiene todos los datos utilizados en la base de datos relacional con su respectivo uso y descripción del dato

Nombre de la base y/o conjunto de datos (Repetir el nombre en todas las filas)	Nombre de la tabla de la base de datos / hoja del conjunto de datos (En caso de que el conjunto de datos no tenga más de una hoja, toma el valor "No aplica")	Etiqueta del campo o variable (atributo)	Descripción	Tipo de dato (Seleccione de acuerdo a las opciones)	Clasificación de la información (Seleccione de acuerdo a las opciones)	Especificaciones y/o reglas de cálculo (En caso de que este elemento no aplique a su base de datos y/o conjunto de datos, registre "No Aplica")	Codificación de valores faltantes (Si este campo no tiene valores faltantes anotar "SIN VALORES FALTANTES")
Restaurante	Personas	IDPersonas	Identificador de cada persona en la tabla	Number	privado	obligatorio	No aplica
Restaurante	Personas	Nombre	Nombre de la persona	String	publico	obligatorio	No aplica
Restaurante	Personas	Prim_Apellido	Primer apellido de la persona	String	publico	obligatorio	No aplica
Restaurante	Personas	Seg_Apellido	Segundo apellido de la persona	String	publico	Opcional	No aplica
Restaurante	Personas	Cedula	Cedula juridica	String	privado	Obligatorio, formato de ingreso de dato 1-111-111	No aplica
Restaurante	Correo	ID_Correo	Identificador de correo de una persona en la tabla	Number	privado	obligatorio	No aplica
Restaurante	Correo	Personas_IDPersona	Llave foranea de la persona a la que	Number	privado	Obligatorio	No aplica
Restaurante	Correo	correo	Correo de la persona	String	privado	Obligatorio	No aplica
Restaurante	Vendedor	ID_Vendedor	Identificado de vendedor de una persona	Number	privado	obligatorio	No aplica
Restaurante	Vendedor	Personas_IDPersona	Llave foranea de la persona de tipo vendedor	Number	privado	obligatorio	No aplica
Restaurante	Vendedor	Usuario	Nombre la cuenta de usuario	String	privado	obligatorio	No aplica
Restaurante	Vendedor	Password	Contraseña de usuario para la aplicacion	Password	privado	obligatorio	No aplica
Restaurante	Cliente	ID_Cliente	Identificador de cliente de una persona	Number	privado	obligatorio	No aplica
Restaurante	Cliente	Personas_IDPersona	Llave foranea de la persona de tipo cliente	Number	privado	obligatorio	No aplica
Restaurante	Venta	Id_Venta	Identificador de la venta en la tabla	Number	publico	obligatorio	No aplica
Restaurante	Venta	Date	Fecha de facturacion	Date	publico	Obligatorio, formato de ingreso de dato	No aplica
Restaurante	Venta	Time	Hora de la facturacion	Time	publico	obligatorio, Ingreso de dato 23:23	No aplica
Restaurante	Venta	Productos	Lista de productos facturados a cobrar	Lista	publico	obligatorio	No aplica
Restaurante	Venta	Vendedor_IDVendedor	Llave foranea del vnededor creador de la	Number	publico	obligatorio	No aplica
Restaurante	Venta	Cliente_IDCliente	Llave foranea del cliente que compra la factura	Number	privado	obligatorio	No aplica
Restaurante	Producto	ID_Producto	Identificador del producto	Number	Publico	Obligatorio	No aplica
Restaurante	Producto	Precio	producto	Double	Publico	Obligatorio	No aplica
Restaurante	Producto	Nombre	Nombre del platillo	String	publico	Obligatorio	No aplica
Restaurante	Producto	Imagen	Ilustracion del platillo	Image	publico	Obliqatorio	No aplica

## Procedimiento almacenado y función dentro de la base de datos de Oracle

Este proyecto incluirá una función en Oracle con la finalidad de sumar el precio de cada producto facturado para dar un total a pagar por el cliente incluyendo el nombre del platillo además de un procedimiento almacenado de la venta contemplando una copia al correo electrónico del cliente para respaldo en caso de que el cliente detecte un error en las cuentas o un producto incorrecto facturado en la cuenta.