



UNIVERSIDAD DE GUADALAJARA  
CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS  
DEPARTAMENTO DE CIENCIAS COMPUTACIONALES

## **Práctica 7 - Diseño Estructural**

Nombre: Oscar Evanilson Gutiérrez Pérez, Código: 219748308

Nombre: Milton Osvaldo Rodriguez González, Código: 216572357

Materia: Seminario de Solución de Problemas de Ingeniería de Software

Sección: D01

Carrera: Ingeniería en Computación

31 de octubre de 2021

# Índice

<b>Introducción</b>	<b>3</b>
<b>Desarrollo y Resultados</b>	<b>5</b>
<b>Conclusiones</b>	<b>11</b>
<b>Referencias</b>	<b>12</b>

## Introducción

A lo largo de las últimas semanas, hemos continuado con el desarrollo y la documentación del proyecto que hemos presentado durante el semestre presente para la empresa "Tabacos Ornelas". En el presente documento realizaremos una serie de diagramas que nos servirán para presentar lo que se conoce como el diseño estructural de nuestro programa.

En nuestro caso, las esquematizaciones que seleccionamos en equipo para realizar nuestro diseño estructural son las siguientes: diagrama de bloques, modelo Entidad-Relación, modelo relacional, tarjetas CRC, diccionario de datos y diagrama de clases.

Las arquitecturas de sistemas se modelan con frecuencia usando diagramas de bloques simples. Cada recuadro en el diagrama representa un componente. Los recuadros dentro de otro recuadro indican que el componente se dividió en subcomponentes. Las flechas significan que los datos y/o señales de control pasan de un componente a otro en la dirección de las flechas.

Los diagramas de bloque presentan una imagen de alto nivel de la estructura del sistema e incluyen fácilmente a individuos de diferentes disciplinas que intervienen en el proceso de desarrollo del sistema (*Unidad III. Diseño Arquitectónico del Software*, s. f.).

Un diagrama entidad-relación, también conocido como modelo entidad relación o ERD, es un tipo de diagrama de flujo que ilustra cómo las "entidades", como personas, objetos o conceptos, se relacionan entre sí dentro de un sistema. Emplean un conjunto definido de símbolos, tales como rectángulos, diamantes, óvalos y líneas de conexión para representar la interconexión de entidades, relaciones y sus atributos (*Qué es un diagrama entidad-relación*, s. f.).

El modelo relacional desarrolla un esquema de base de datos (*database schema*) a partir del cual se podrá realizar el modelo físico o de implementación en el DBMS.

Este modelo está basado en que todos los datos están almacenados en tablas (entidades/relaciones) y cada una de estas es un conjunto de datos, por tanto una base de datos es un conjunto de relaciones. La agrupación se origina en la tabla: tabla -> fila (tupla) -> campo (atributo) (Araneda, s. f.).

Las tarjetas CRC se utilizan, individualmente, para representar objetos. La clase del objeto puede ser escrita en la parte superior de la tarjeta, las responsabilidades enumeradas en la parte izquierda y las clases que colaboran son listadas a la derecha de cada responsabilidad (Qué Son y para Qué Sirven Las Tarjetas CRCs | PDF, s. f.).

Un diccionario de datos es un tipo de metadato que enlista de manera organizada los nombres, definiciones y características de cada uno de los campos o atributos de una base de datos y/o conjunto de datos (Elaboración de Diccionarios de Datos, s. f.).

El diagrama de clases recoge las clases de objetos y sus asociaciones. En este diagrama se representa la estructura y el comportamiento de cada uno de los objetos del sistema y sus relaciones con los demás objetos, pero no muestra información temporal.

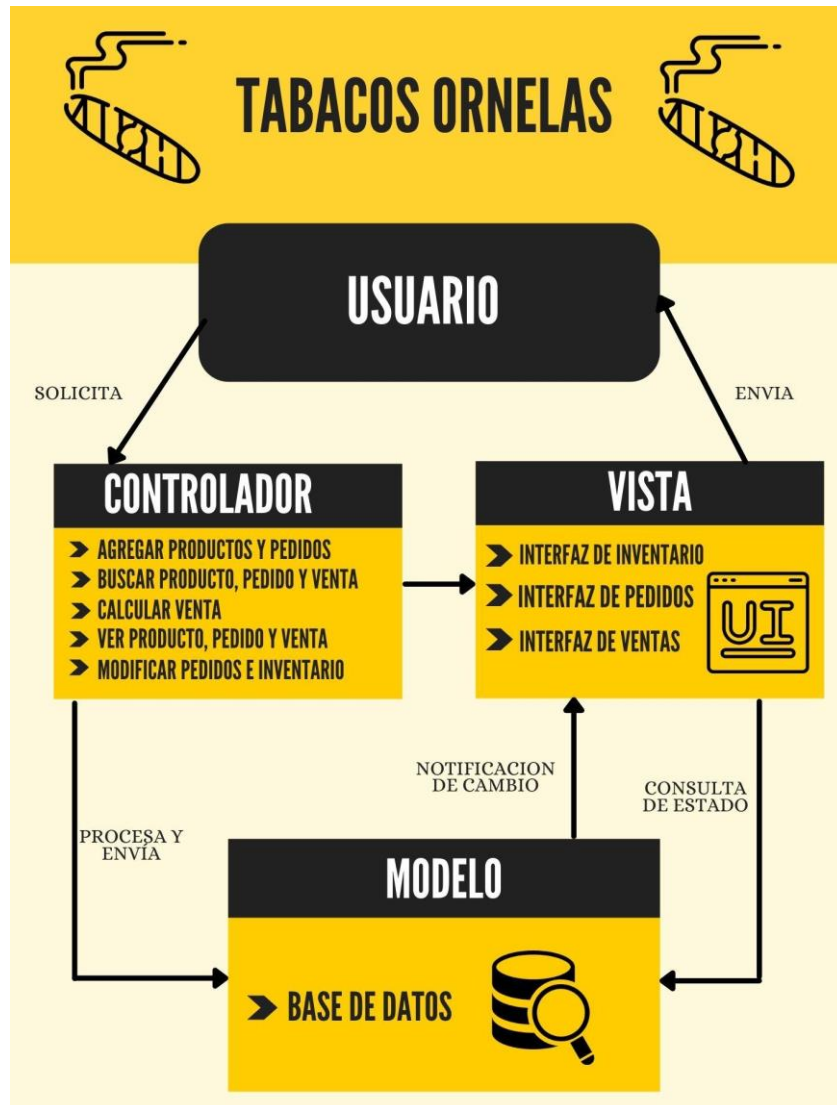
El objetivo principal de este modelo es la representación de los aspectos estáticos del sistema, utilizando diversos mecanismos de abstracción (clasificación, generalización, agregación) (*Diagrama de Clases - manuel.cillero.es*, s. f.).

Con todo lo anterior mencionado, damos paso a la presentación de los diagramas que realizamos y del diseño estructural de nuestro programa administrador para la empresa de tabacos.

## Desarrollo y Resultados

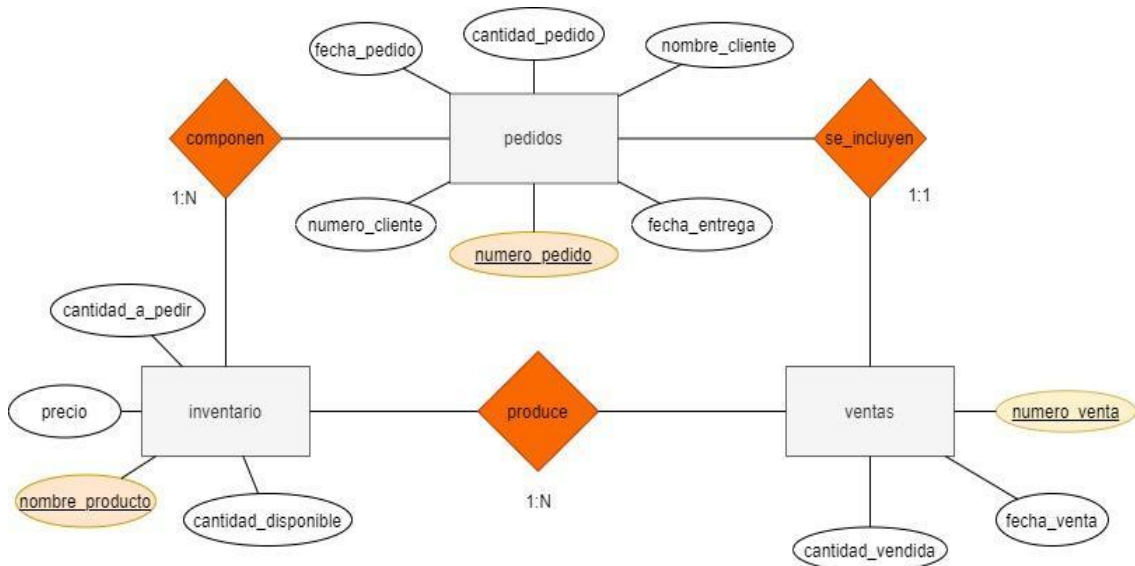
### Diagrama de bloques:

Los diagramas de bloque son una forma adecuada para describir la arquitectura del sistema durante el proceso de diseño, pues son una buena manera de soportar las comunicaciones entre las personas involucradas en el proceso.



En este diagrama decidimos realizar nuestro programa en una arquitectura MVC, donde podemos ver que en el modelo se encontrará la base de datos y que gracias a el controlador, donde se realizarán las tareas y procesos del programa, el usuario podrá acceder al sistema por medio de una vista de interfaces, una de ventas, de pedidos y de inventario.

## Diagrama Entidad-Relación:



En nuestro caso, definimos tres entidades representadas con rectángulos: la de ventas, inventario y pedidos, que tienen sus atributos representados con óvalos. Dentro de estos atributos encontramos las llaves primarias, marcadas con otro color y subrayadas ya que estos atributos no se pueden repetir y nos sirven para identificar los datos que ingresamos. También, podemos ver las relaciones que comparten las tres entidades, estas relaciones están representadas con rombos y a un lado de estas, se encuentran las cardinalidades de las mismas y así saber cómo se manejan estas relaciones, podemos ver que los pedidos se componen de un inventario, así mismo este inventario produce muchas ventas y por último un pedido se incluye en una venta.

## Modelo Relacional:

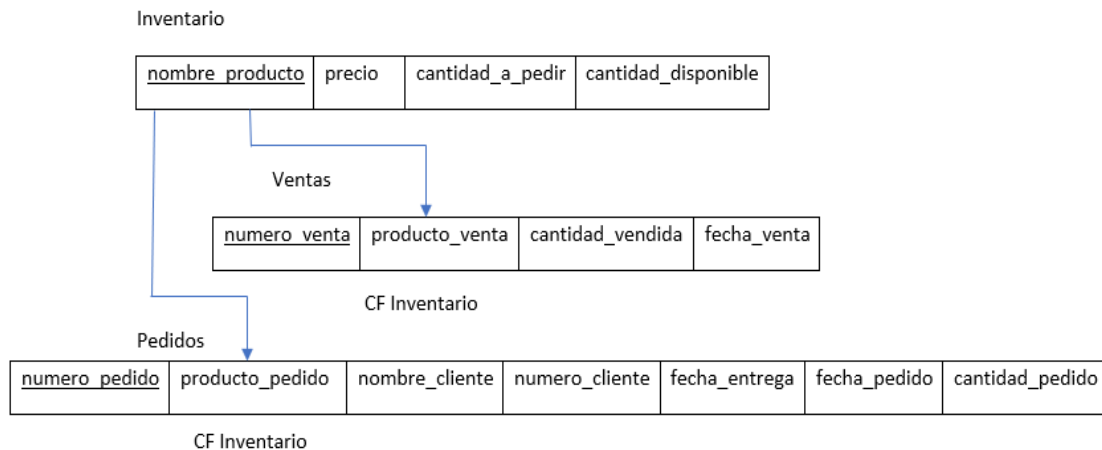
*Modelo conceptual:*

Inventario=(nombre\_producto, precio, cantidad\_a\_pedir, cantidad\_disponible)

Ventas=(numero\_venta, producto\_venta, cantidad\_vendida, fecha\_venta)

Pedidos=(numero\_pedido, producto\_pedido, nombre\_cliente, numero\_cliente, fecha\_entrega, fecha\_pedido, cantidad\_pedido)

### Modelo relacional:



En este caso nuestro modelo relacional es la representación en tablas de nuestro modelo entidad-relación. Agregamos las claves foráneas (que no estaban presentes como atributos en el diagrama anterior, sino que estaban implícitos en las cardinalidades y las relaciones). Nuestro modelo relacional terminó siendo muy pequeño, únicamente contamos con tres tablas, las llaves foráneas se mantienen. Las flechas indican que el campo nombre\_producto se convierte en clave foránea en las otras tablas con el nombre del campo que señala la flecha. En este diagrama, los atributos se vuelven campos.

### Tarjetas CRC:

Inventario	
<ul style="list-style-type: none"> <li>- Agregar productos</li> <li>- Buscar productos</li> <li>- Modificar productos</li> <li>- Ver productos</li> </ul>	Productos Inventario

Ventas	
<ul style="list-style-type: none"> <li>- Buscar venta</li> <li>- Ver ventas</li> <li>- Calcular venta</li> </ul>	Ventas Productos

Pedidos	
<ul style="list-style-type: none"> <li>- Agregar pedido</li> <li>- Buscar pedido</li> <li>- Ver pedidos</li> <li>- Modificar pedidos</li> </ul>	Pedidos Productos

Ui_MainWindow	
<ul style="list-style-type: none"> <li>- Muestra las opciones para el usuario</li> <li>- Interfaz de inventario</li> <li>- Interfaz de ventas</li> <li>- Interfaz de pedidos</li> <li>- Obtiene los datos que el usuario ingresa</li> <li>- Muestra la salida del programa</li> </ul>	Inventario Ventas Pedidos Entrada de usuario Usuario

Mainwindow	
<ul style="list-style-type: none"> <li>- Conecta la interfaz con la base de datos</li> <li>- Guarda los datos ingresados por el usuario</li> <li>- Pasa los datos que se muestran en la interfaz</li> <li>- Realiza las funciones que solicita el usuario en interfaz</li> <li>- Funciones de ventas, pedidos e inventario</li> </ul>	Base de Datos Entrada de usuario Usuario Ventas Pedidos Inventario Ui_MainWindow

En las tarjetas CRC dividimos nuestro programa en clases, cada clase en una tarjeta, la cual tiene su nombre en la parte de arriba de la tarjeta.

La parte inferior de cada tarjeta está dividida en dos, la izquierda contiene las tareas y funciones que puede realizar cada clase y la derecha los objetos que manipula dicha clase.

En nuestro caso elegimos una por cada entidad: ventas, inventario y pedidos, que administrarán cada objeto y las funciones de estas entidades; además seleccionamos una clase que se encargará de las interfaces y la salida del programa, así como una última clase de la parte trasera del sistema, que conecta la base de datos con las interfaces y controla el flujo de nuestro programa.



## Diccionarios de Datos:

<b>Nombre de la tabla</b>	<b>Inventario</b>	<b>Fecha de creación:</b>	<b>27/10/2021</b>
<b>Descripción:</b>	<b>Tabla que almacena datos de productos en el inventario.</b>		

Nombre del campo	Tipo de dato	Tamaño	Valor por Default	Fórmula	Valor nulo	Descripción
nombre_producto	varchar	50	"Desconocido"	-	No	PK
precio	money	-	"0.00"	-	No	-
cantidad_a_pedir	integer	4 bytes	0	-	No	-
cantidad_disponible	integer	4 bytes	0	-	No	-

En el siguiente diccionario de datos mostramos información diversa respecto a los campos de nuestras tablas, tales como el tamaño, el valor por default, etc. En la parte de descripción únicamente señalamos si es llave primaria o foránea. En este diccionario no existen valores nulos ni fórmulas.

<b>Nombre de la tabla</b>	<b>Ventas</b>	<b>Fecha de creación:</b>	<b>27/10/2021</b>
<b>Descripción:</b>	<b>Tabla que almacena datos de las ventas completadas.</b>		

Nombre del campo	Tipo de dato	Tamaño	Valor por Default	Fórmula	Valor nulo	Descripción
numero_venta	serial	1 a 21474836	-	-	No	PK
producto_venta	varchar	50	"Desconocido"	-	No	FK
cantidad_vendida	integer	4 bytes	0	-	No	-
fecha_venta	date	-	-	-	No	-

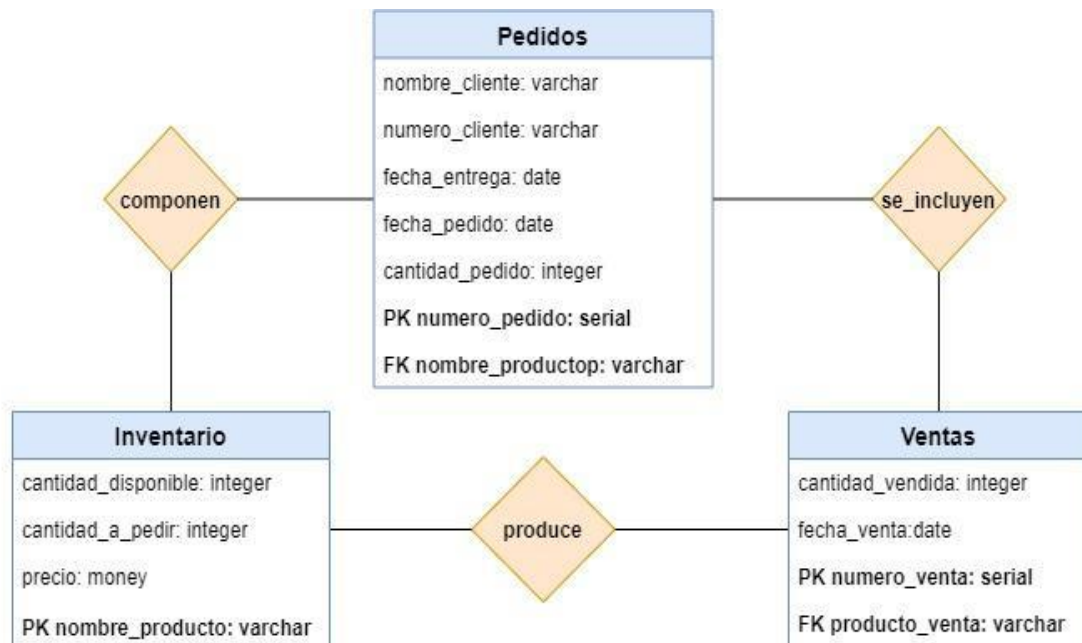
En este diccionario agregamos un tipo de dato serial, que es el numero\_venta, esto para que sea un número único de venta y así también considerarlo como llave primaria. El producto\_venta aparece como llave foránea, proveniente de la tabla de inventario, por lo que todas las características de ese campo se conservan. En esta tabla tampoco se incluyen fórmulas ni valores nulos.

<b>Nombre de la tabla</b>	<b>Pedidos</b>	<b>Fecha de creación:</b>	<b>27/10/2021</b>
<b>Descripción:</b>	<b>Tabla que almacena datos de los pedidos de productos.</b>		

Nombre del campo	Tipo de dato	Tamaño	Valor por Default	Fórmula	Valor nulo	Descripción
numero_pedido	serial	1 a 21474836	-	-	No	PK
producto_pedido	varchar	50	"Desconocido"	-	No	FK
nombre_cliente	varchar	60	"Desconocido"	-	No	-
numero_cliente	varchar	10	"Desconocido"	-	No	-
fecha_entrega	date	-	-	-	Sí	-
fecha_pedido	date	-	-	-	No	-
cantidad_pedido	integer	4 bytes	0	-	No	-

En este último diccionario, nuevamente incluimos como tipo serial a un campo, que es el numero\_pedido, el cual será la clave primaria de la tabla. El producto\_pedido será la llave foránea, proveniente de la tabla de inventario, por lo que conserva sus características. En este caso tenemos un campo que puede tener valor nulo, el cual es la fecha\_entrega, lo consideramos así porque puede que no exista una fecha específica, sino que dependa de la disponibilidad del cliente.

## Diagrama de clases:



En este diagrama, tenemos algo muy similar a lo que presentamos en el entidad-relación, ahora vemos las entidades en recuadros que debajo de sus nombres contienen sus atributos, en negrita podemos ver las llaves foráneas y las llaves primarias de cada entidad y por último en los rombos seguimos viendo las relaciones que comparten dichas entidades.

## Conclusiones

Esta actividad fue la más larga y laboriosa que hemos realizado para nuestro proyecto, fue interesante investigar acerca de los diferentes diagramas que nos ayudan a representar una visualización cercana a lo que al final será el proyecto que entregaremos y espero que sea de mayor facilidad construirlo después de tanta documentación que hemos realizado. En conclusión, esta actividad me dejó una mejor idea de cómo se va a ver nuestro programa internamente y de cómo se va a relacionar para que pueda tener un buen funcionamiento y cumpla las necesidades del usuario al momento de presentárselo a los dueños de la empresa con la que estamos colaborando. Espero que con la ayuda que hemos desarrollado con la documentación que llevamos realizando desde hace algunas semanas podamos tener un mejor proyecto a comparación del que teníamos pensado al inicio de semestre.

## Referencias

1. Araneda, P. (s. f.). *Capítulo 7 El Modelo Relacional | Base de Datos*. Recuperado 27 de octubre de 2021, de <https://bookdown.org/paranedagarcia/database/el-modelo-relacional.html>
2. *Diagrama de Clases—Manuel.cillero.es*. (s. f.). Recuperado 28 de octubre de 2021, de <https://manuel.cillero.es/doc/metodologia/metrica-3/tecnicas/diagrama-de-clases/>
3. *Elaboración de diccionarios de datos*. (s. f.). Portal de La Política de Datos de La Ciudad de México. Recuperado 27 de octubre de 2021, de <https://politicadedatos.cdmx.gob.mx/cultura/guias/diccionario>
4. *Qué es un diagrama entidad-relación*. (s. f.). Lucidchart. Recuperado 28 de octubre de 2021, de <https://www.lucidchart.com/pages/es/que-es-un-diagrama-entidad-relacion>
5. *Qué Son y para Qué Sirven Las Tarjetas CRCs | PDF*. (s. f.). Scribd. Recuperado 28 de octubre de 2021, de <https://es.scribd.com/document/207429482/Que-son-y-para-que-sirven-las-tarjetas-CRCs-docx>
6. *Unidad III.Diseño Arquitectonico del Software*. (s. f.). Recuperado 27 de octubre de 2021, de <https://classroom.google.com/u/1/c/Mzc1NTM2Njl2ODg3/m/NDA2NzQ3Njc3OTE3/details>