



Lenguajes de Programación II

Programación Orientada a Objetos

UNIDAD IV – PARTE 2

DISEÑO ORIENTADO A OBJETOS

Docente: Lic. Rodrigo Velázquez

FACULTAD POLITÉCNICA – UNA

2022



DOO



CONTENIDO

Diseño Orientado a Objetos

- Diseño
- Diagramas de Interacción
- Diagramas de secuencia
- Diagramas de clases
- Diagrama de Base de Datos

DISEÑO

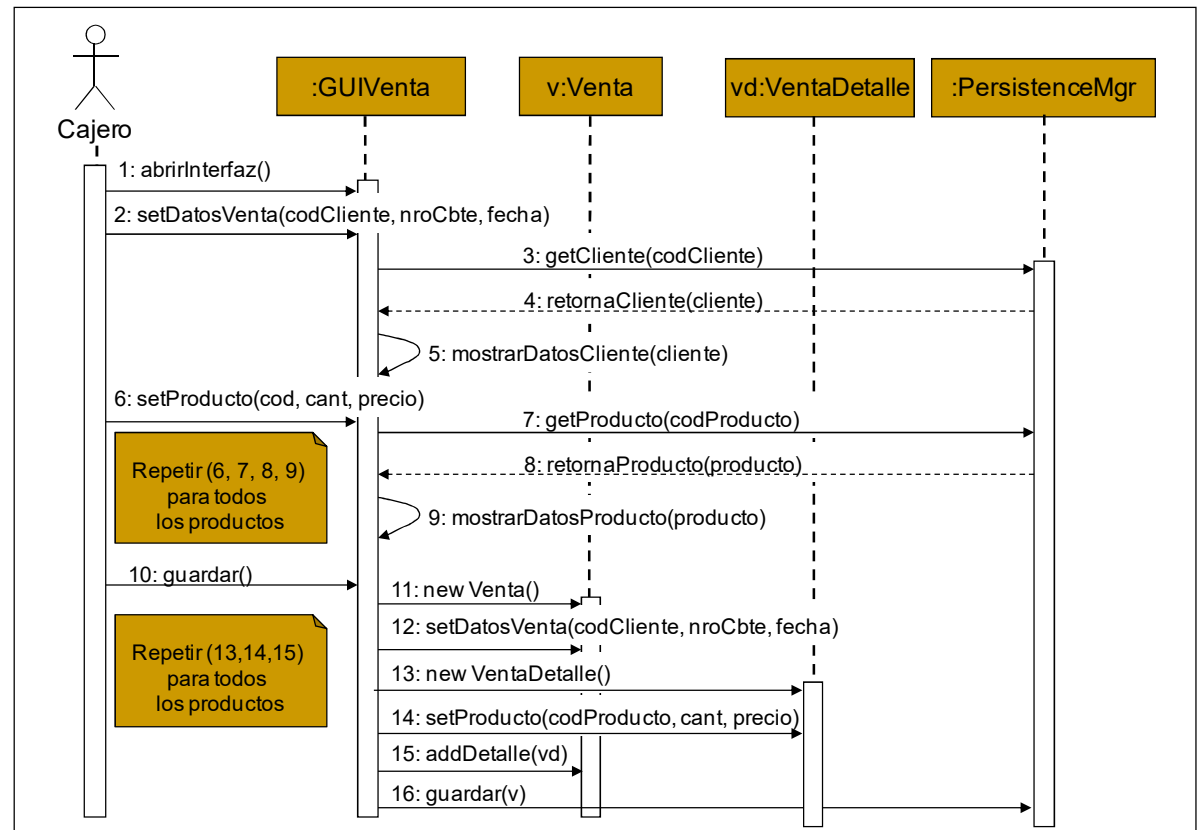
- Es la etapa en donde se idea el esquema lógico para la construcción de un Software.
- Algunas de las tareas que se llevan a cabo durante la fase de diseño de un Software son las siguientes:
 - Definir los diagramas de interacción.
 - Diagramas de secuencia *
 - Diagramas de colaboración.
 - Definir los diagramas de diseño de clases.
 - Definir el esquema de la base de datos.
 - Definir los reportes, la interfaz de usuario y la secuencia de las pantallas.

DIAGRAMAS DE INTERACCIÓN

- Su función es explicar gráficamente como los objetos interactúan a través de mensajes para realizar las tareas del sistema definidas por los casos de uso.
- En UML existen dos diagramas que muestran las interacciones entre los objetos.
 - Diagramas de Secuencia
 - Diagramas de Colaboración

DIAGRAMAS DE SECUENCIA

- Los Diagramas de Secuencia muestra las interacciones entre un grupo de objetos y los actores en un formato de línea temporal.
- Diagrama de Secuencia para el caso de uso “Venta de Productos”:



DIAGRAMAS DE INTERACCIÓN

- Los diagramas de interacción deben estar acompañados de una descripción que explique que ocurre en cada paso.
- En el caso anterior la explicación sería:
 - El usuario ingresa a la interfaz de carga de ventas.
 - El usuario ingresa los datos de la venta (fecha, numero de comprobante, código del cliente).
 - El sistema recupera y presenta la información del cliente (nombre, dirección, RUC).
 - Por cada producto vendido:
 - El usuario ingresa el código del producto vendido y el sistema recupera y presenta su descripción y precio.
 - El usuario ingresa la cantidad y el precio final de venta
 - El sistema actualiza el monto de la venta.
 - Una vez cargados todos los datos, el usuario indica que la venta se debe guardar.
 - El sistema crea y carga los objetos que representan la venta y guarda los datos en el repositorio de almacenamiento persistente.
 - La interfaz se limpia, permitiendo la carga de una nueva venta.

DIAGRAMAS DE INTERACCIÓN

- En la fase de creación de los diagramas de interacción se determinan otros objetos (aparte de los determinados por los conceptos) que brindan funcionalidades complementarias al sistema
- Ejemplos de estos Objetos son:
 - Objetos de Interfaz: objetos que implementan las interfaces de usuario.
 - Objetos de Datos: objetos que brindan las funcionalidades que implementan el mecanismo de persistencia de los datos.

DIAGRAMA DE CLASES

- Un diagrama de clases indica las relaciones que existen entre las clases que involucran la implementación del sistema.
- Un diagrama de clases esta compuesto de:
 - **Clases:** indicando sus atributos y métodos, y la visibilidad de ellos.
 - **Relaciones:** indicando los casos de *Herencia*, *Composición*, *Agregación*, *Asociación* y *Uso*.
- El diagrama de clases se crea a partir de:
 - El modelo conceptual, identificando los conceptos que deberán ser implementados por el sistema.
 - Los diagramas de interacción, a partir de lo cuales se determina los métodos y atributos que serán parte de las definiciones de las clases.

DIAGRAMAS DE CLASES

- Las clases son representadas por cuadros que indican:
 - El nombre de la Clase
 - En formato normal cuando la clase es efectiva
 - En formato de cursiva cuando la clase es abstracta
 - Agregando el estereotipo <<interface>> cuando es una interfase.
 - Los Atributos, indicando:
 - La visibilidad, (+) publico, (#) protegido, (-) privado.
 - El nombre
 - El tipo
 - Los Métodos, indicando:
 - La visibilidad, igual que con los atributos,
 - El nombre, si es efectivo o abstracto según formato
 - El nombre, tipo y orden de los parámetros,
 - El tipo de retorno.

DIAGRAMA DE CLASES

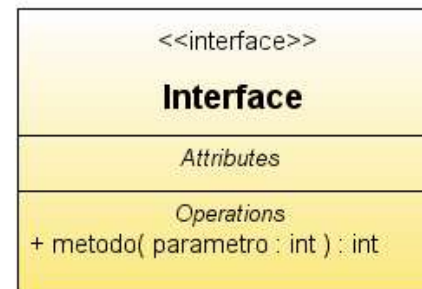
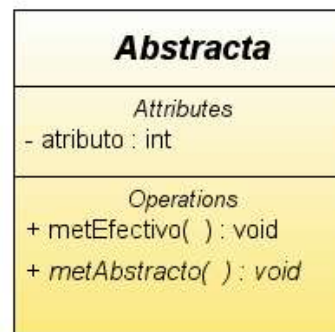
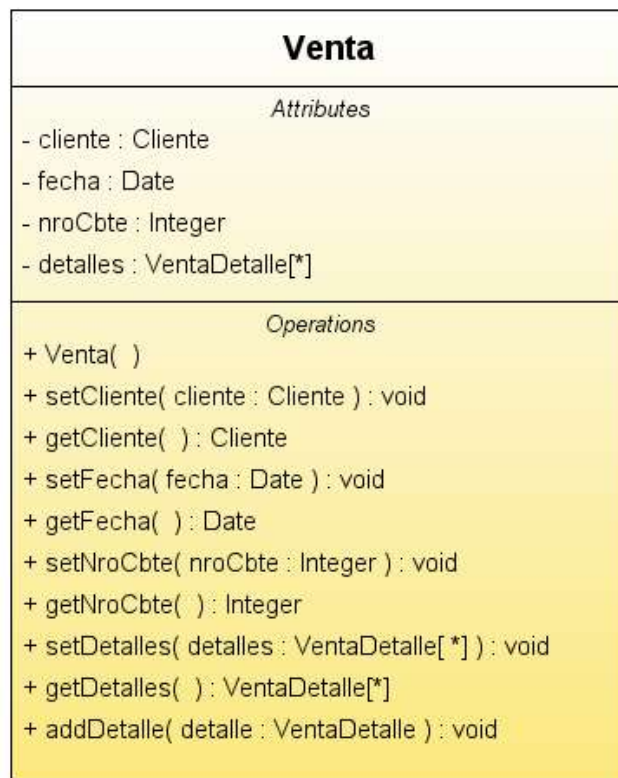
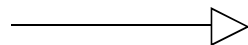


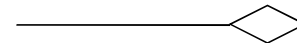
DIAGRAMA DE CLASES

- Las relaciones entre las clases son la siguientes:
 - **Herencia:** indicando que una clase hereda de otra o implementa un interfase.
 - **Uso o Asociación:** indica que una clase hace uso de otra en su implementación.
 - **Agregación:** indica que un objeto de una clase incluye como atributo una o mas referencias a objetos de otra clase, pero el tiempo de vida de estos es independiente
 - **Composición:** indica que un objeto de una clase se incluye como atributo una o mas referencias a objetos de otra clase, donde el tiempo de vida de los objetos incluidos esta determinado por el tiempo de vida del objeto incluyente.
 - **Dependencia o Instanciación:** indica que un objeto de una clase instancia un objeto de otra clase.
- Las relaciones entre clases por lo general también indican una cardinalidad de relacionamiento.

DIAGRAMAS DE CLASES



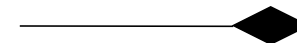
HERENCIA



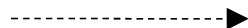
AGREGACIÓN



USO



COMPOSICIÓN



INSTANCIACIÓN

DIAGRAMA DE CLASES

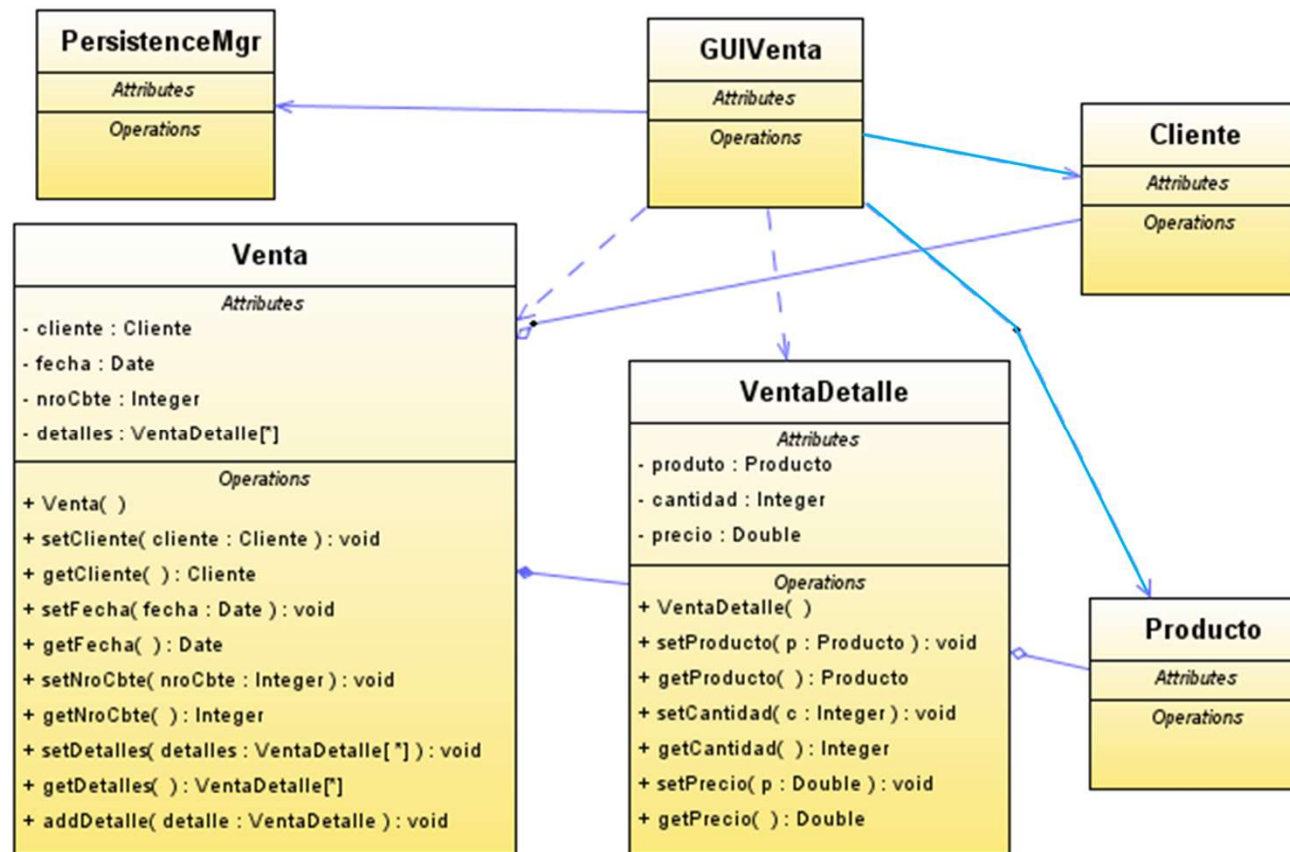


DIAGRAMA DE BASE DE DATOS

- El objetivo de este diagrama es mostrar como los objetos del negocio son mapeados a un esquema de Base de Datos Relacional
- La conversión más fácil consiste en hacer corresponder las clases con tablas.

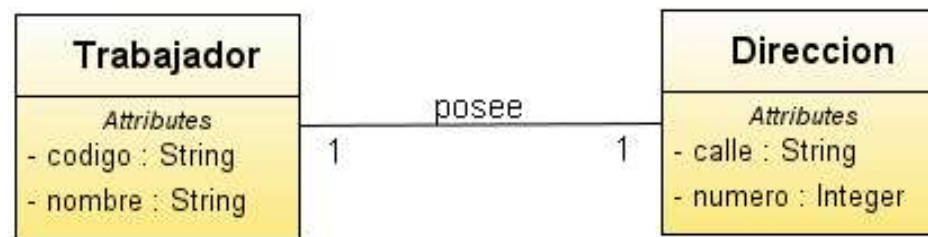
Cliente
<i>Attributes</i>
- codigo : String
- nombre : String
- ruc : String
- direccion : String
- telefono : String

Cliente
codigo Varchar(n)(10) NN (PK)
nombre Varchar(n)(50) UNN
ruc Varchar(n)(15) UNN
direccion Varchar(n)(200) UNN
telefono Varchar(n)(20)

- La tabla correspondiente a una Clase del negocio será responsable de almacenar todas las instancias de dicha Clase.

DIAGRAMA DE BASE DE DATOS

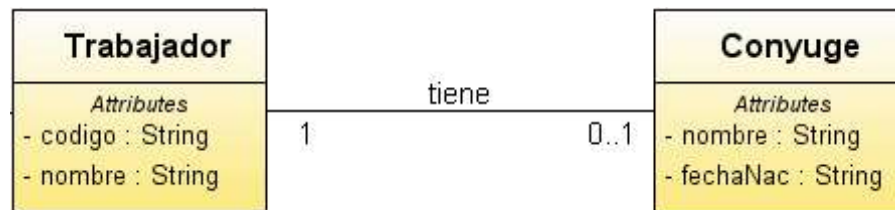
- Cuando la relación es **1 a 1**, la existencia de un objeto esta determinada por la existencia del otro.
- En este caso ambas clases se mapean en una única tabla.



Trabajador				
codigo	Serial	NN	(PK)	
nombre	Char(20)	NN		
calle	Char(20)	NN		
numero	Char(20)	NN		

DIAGRAMA DE BASE DE DATOS

- Cuando se tiene una relación de **1 a 0..1**:
 - Las clases pueden ser mapeadas en una única tabla en donde los atributos de la clase que puede o no existir pueden ser nulos



Trabajador			
codigo	Serial	NN	(PK)
nombre	Char(20)	NN	
nombre_conyuge	Char(20)		
fec_nac_conyuge	Char(20)		

DIAGRAMA DE BASE DE DATOS

- Cuando se tiene una relación de **1 a 0..1**:
 - O cada Clase puede ser mapeada en una tabla, en donde la tabla correspondiente a la clase que puede o no existir posee una clave foránea a la otra tabla.

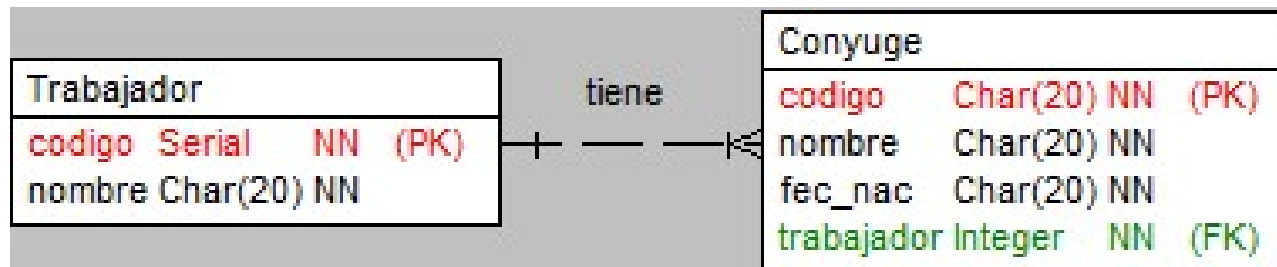
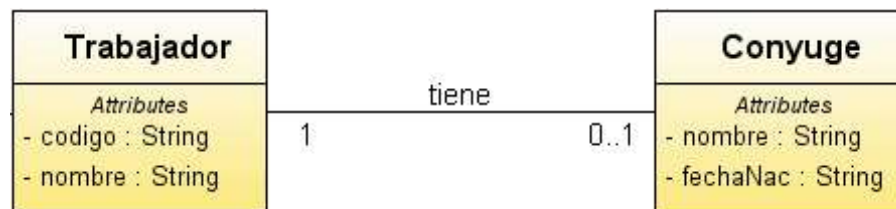
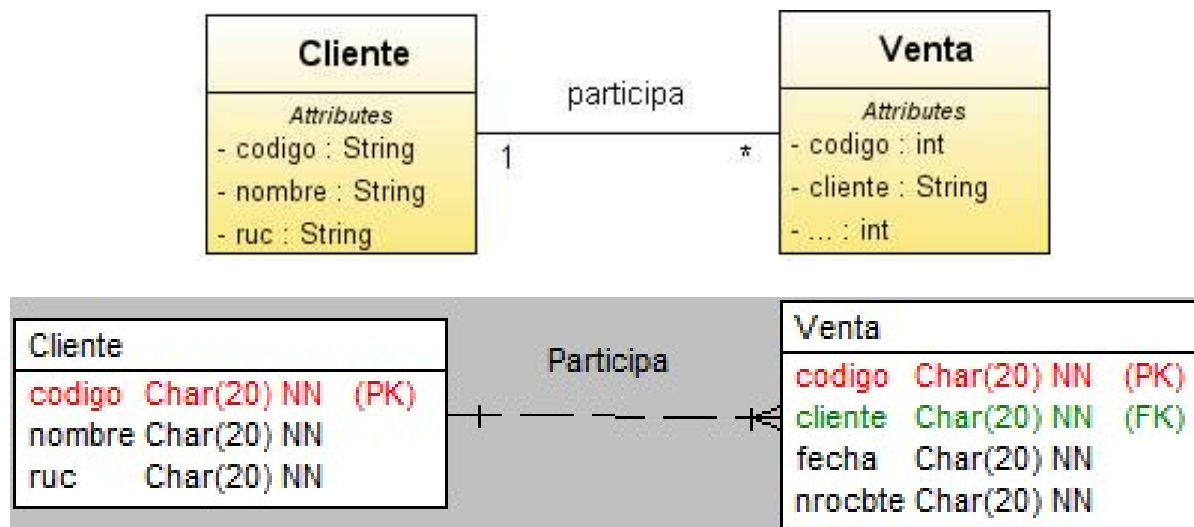


DIAGRAMA DE BASE DE DATOS

- Cuando existe una relación de **1 a muchos** (*, 1..*, 0..20) las clases se mapean en tablas diferentes, en donde la tabla correspondiente a la multiplicidad **muchos** posee una clave foránea a la otra tabla.



DIAGRAMAS DE BASE DE DATOS

- Cuando la relación en **muchos a muchos**, cada Clase se mapea en una tabla y para la relación se crea una tabla que posee una clave foránea a cada tabla

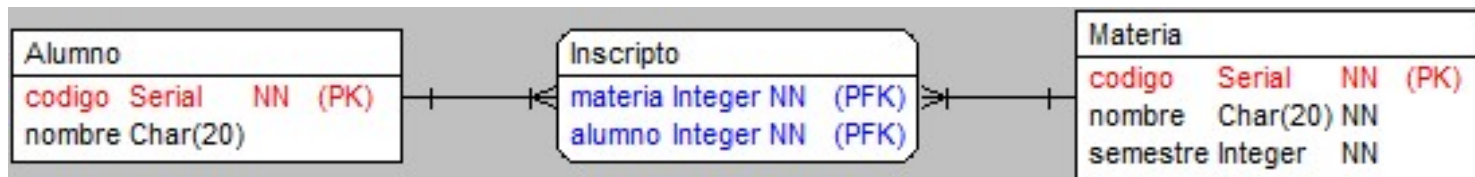
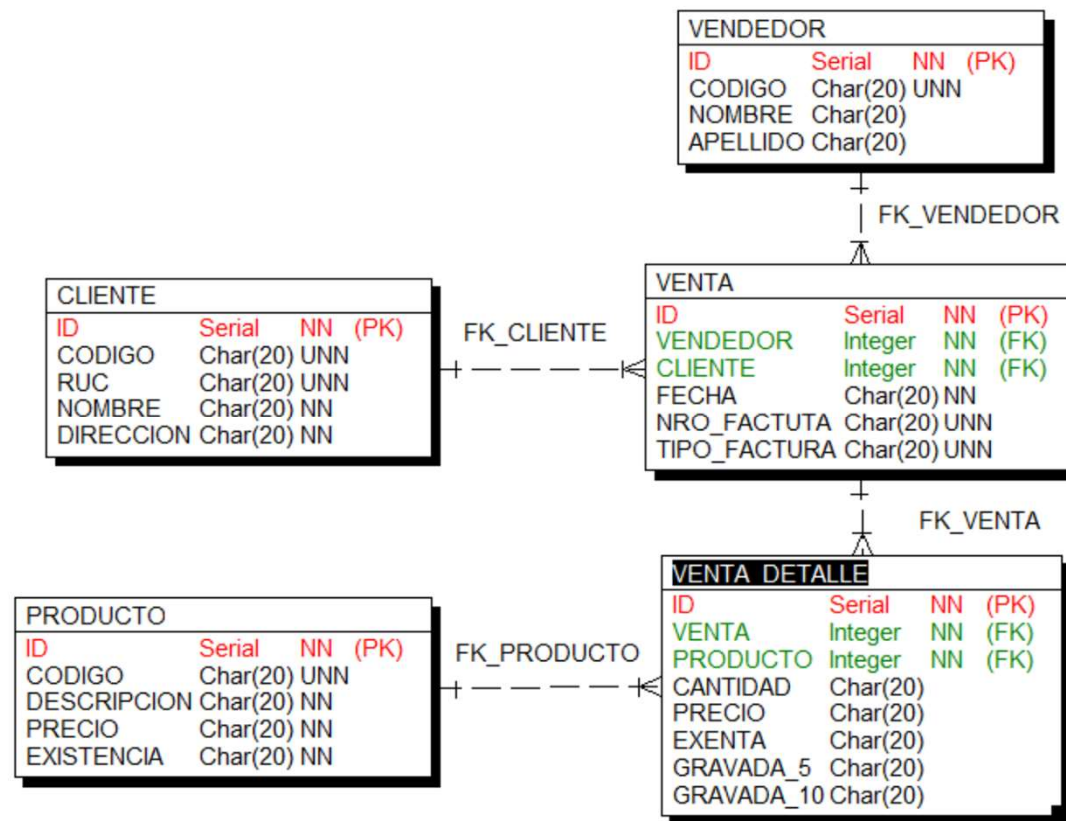
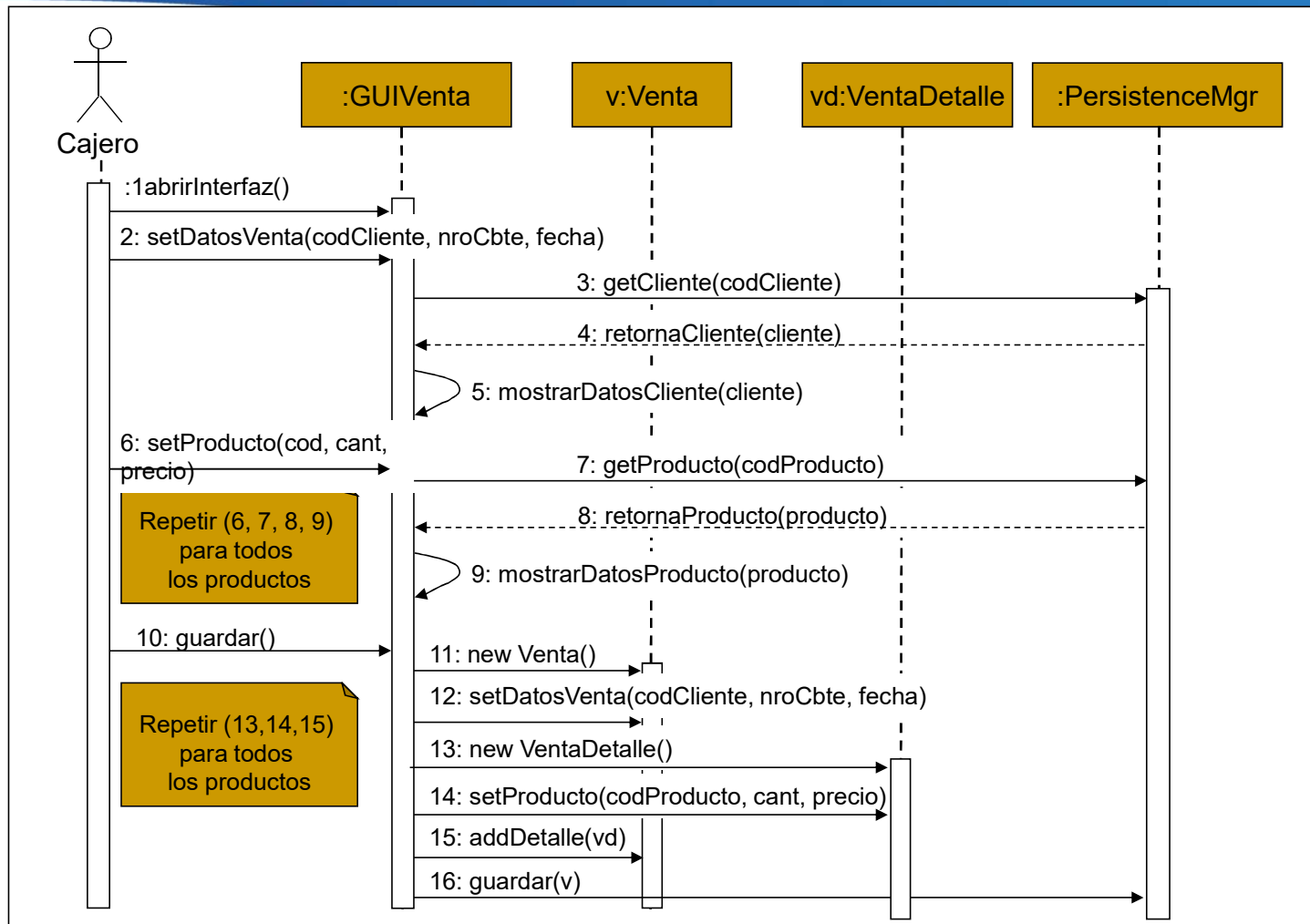


DIAGRAMA DE BASE DE DATOS







BIBLIOGRAFÍA

Deitel, P. y Deitel, H. (2016), Cómo programar en Java. Décima edición. México: Pearson educación.

FIN!

