

Control uno

Fundamentos de Ingeniería del Software.

ÍNDICE

1. MODELO CONCEPTUAL

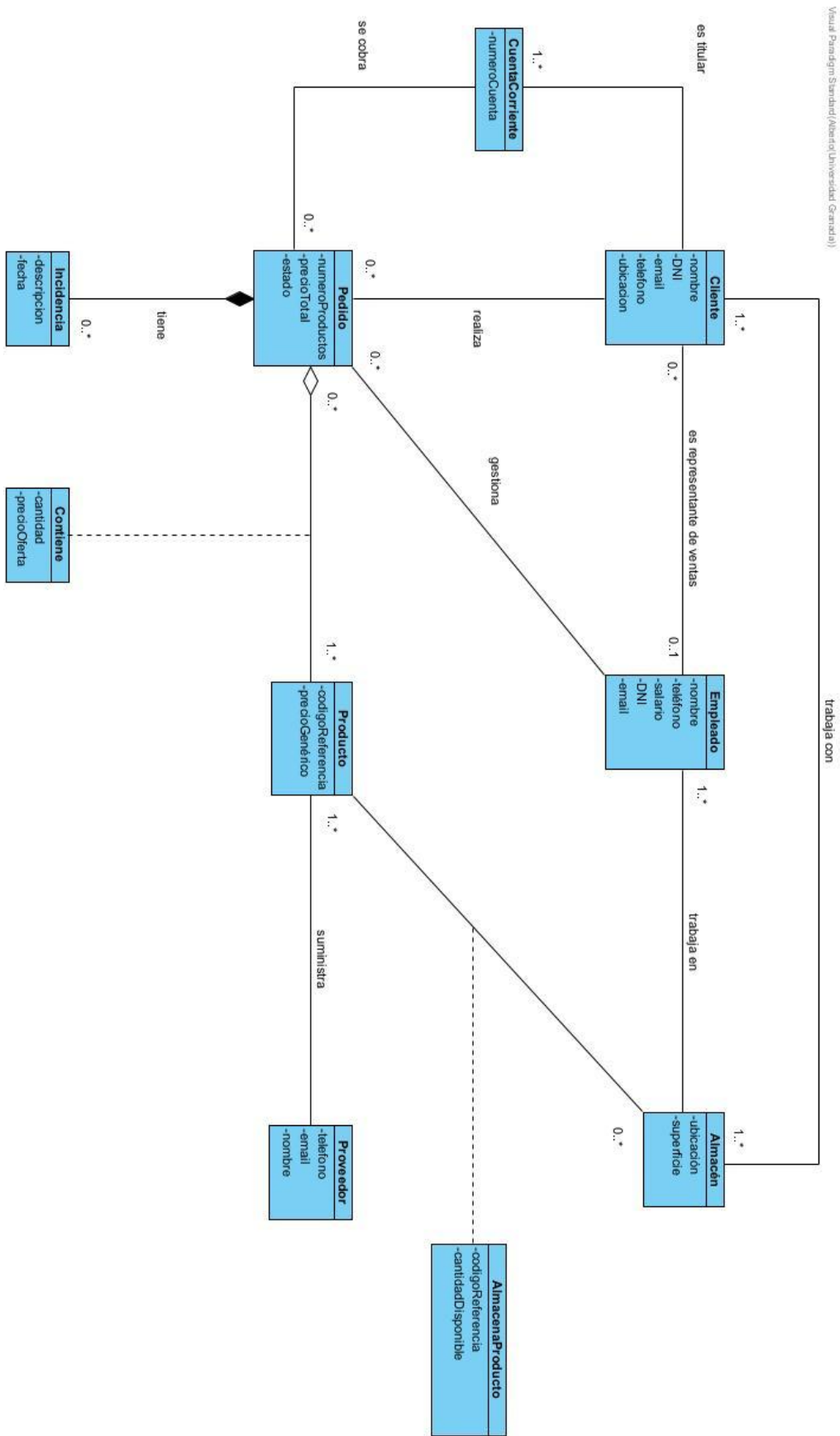
1. Diagrama
2. Justificación de decisiones importantes

2. CONTRATOS

1. añadirListaEspera
2. añadirPedido

1. MODELO CONCEPTUAL

1. Diagrama



2. Justificación de decisiones importantes

- He creado la clase Producto para relacionarla con la clase Pedido mediante una agregación (ya que un pedido está formado por una serie de productos, pero si el pedido deja de existir, los productos no desaparecen).
- Para representar los datos que se almacenan sobre cada producto almacenado en cada almacén, he creado una clase AlmacenaProducto que registra el precio del producto y qué cantidad disponible hay de este en el almacén.
- Para representar la cantidad de cada producto y el precio que cuesta en el pedido asociado (precioOferta), he creado la clase Contiene.
- Para las incidencias, he creado una clase Incidencia que está asociada mediante una composición a Pedido (esto se debe a que una incidencia no existe sin un pedido al que estar asociada).
- También quiero señalar que la cardinalidad en “es representante de ventas” es 0..1 porque en el enunciado se nos dice que “normalmente” un cliente tiene un representante de ventas, lo cual yo lo interpreto como que puede tenerlo o no.

2. CONTRATOS

1. añadirListaEspera

Nombre	añadirListaEspera(idRestaurante,idCliente,fecha,hora)
Responsabilidad	Añade a la lista de espera del restaurante con idRestaurante una ReservaEnCola con la fecha y hora indicados del cliente con idCliente
Tipo	Sistema(Restaurante)
Notas	No se pueden aceptar reservas en cola que poseen fecha y hora de la anteriores o iguales al momento en el que se hace la reserva
Excepciones	<ul style="list-style-type: none">• idRestaurante no existe o no tiene un formato correcto• idCliente no existe o no tiene un formato correcto• La fecha y hora no tienen un formato correcto• La fecha y hora indicadas son anteriores al momento actual
Salida	Devuelve un booleano que indica si ha habido éxito
Precondiciones	
Postcondiciones	<ul style="list-style-type: none">• Fue creado un objeto, reserva, de la clase ReservaEnCola debidamente inicializado con los datos aportados• El objeto reserva se ha añadido a la lista de espera del restaurante identificado por idRestaurante

2. añadirPedido

Nombre	añadirPedido(idRestaurante,NumeroMesa,listaIdConsumición)
Responsabilidad	Añade un pedido a una mesa de un restaurante
Tipo	Sistema(Restaurante)
Notas	
Excepciones	<ul style="list-style-type: none"> • idRestaurante no existe o no tiene un formato correcto • NumeroMesa es mayor que el número de mesa más alto del restaurante o tiene un formato incorrecto • El idConsumicion de algún objeto de la lista de consumiciones que componen el pedido no existe
Salida	Devuelve un booleano que indica si ha habido éxito
Precondiciones	
Postcondiciones	<ul style="list-style-type: none"> • Fue creado un objeto, pedido, de la clase Pedido que está compuesto por objetos de la clase Consumición cuyos ID's se corresponden con los de listaIdConsumición. • Fue creado un enlace entre pedido y el objeto Mesa, identificado por NumeroMesa que coincide con el número que era parámetro de la operación.