ESCUELA SUPERIOR DE INGENIERÍA INFORMÁTICA - UNIVERSIDAD DE VIGO

Interfaces de Usuario

Entrega 2

Grupo 2

22/11/2013

GRUPO A:

Líder de equipo y proyecto: Adrián Célix Fernández
Carmen Selina Meire Pérez
Laura Lorenzo Pérez
Daniel Pavón Llamas

GRUPO B:

Líder de equipo: Alberto Gutiérrez Jácome
Daniel Sánchez Valencia
Daniel Álvarez Outerelo
Marcos Núñez Celeiro
David Lorenzo Dacal

Contenido

Contenido 1
Entorno 4
Comunicación y estructura de desarrollo 4
Arquitectura de desarrollo 4
Capa de acceso a datos 4
Modelo 5
Controlador 5
Vista 6
Normas de estandarización 6
Base de datos 6
Código PHP 6
Sistema de comunicación y control de cambios 8
Diagrama de casos de uso 8
Prototipado falso 9
ChangeLog 9
Sistema de comunicación entre los miembros del grupo 10
Diagramas de secuencia detallados 11
Registrar usuario 11
Login usuario 11
Consultar usuario 12
Modificar usuario 12
Eliminar usuario 13
Añadir artículo 13
Buscar artículos 14
Listar artículos 14
Modificar artículo 15
Eliminar artículo 15
Añadir venta 16
Listar ventas 16
Modificar venta 17
Eliminar venta 17

Añadir subasta 18
Listar subasta 18
Eliminar subasta 19
Pujar 19
Comprar 20
Realizar pago 20
Calificar venta 21
Cambiar porcentajes 21
Diagramas de clases parciales 22
Registrar usuario 22
Login usuario 22
Consultar usuario 23
Modificar usuario 23
Eliminar usuario 24
Añadir artículo 24
Listar artículos 25
Buscar artículo 25
Modificar artículo 26
Eliminar artículo 26
Añadir venta 27
Buscar venta 27
Listar ventas 28
Modificar venta 28
Eliminar venta 29
Añadir subasta 29
Listar subastas 30
Eliminar subastas 30
Pujar 31
Comprar 31
Realizar pago 32
Calificar venta 32
Cambiar porcentajes 33
Diagrama de clases total 34
Documentación BD 35

Diagrama Entidad-Relación 35
Diagrama de datos de la BD 36
Estructura física 37
Diccionario de datos 39

Entorno

Comunicación y estructura de desarrollo

Para la realización del proyecto se ha realizado un repositorio en git que se ha dividido en dos branches (ramas).

- **Rama master:** La rama master sirve para realizar modificaciones finales, versiones más antiguas y menos modificadas pero siempre totalmente funcionales.

- **Rama develop:** La rama develop es donde se realizan todos los cambios. Cada cambio realizado por un miembro de alguno de los dos equipos de proyecto se ha ido subiendo a git, siempre y cuando fuese ya un cambio completo (una funcionalidad).

La estructura de desarrollo se ha realizado, primero dividiendo el grupo en dos equipos y separando así la realización de la base de datos de la documentación relacionada con la arquitectura del sistema (Diagramas de clases y secuencia) en la primera parte de esta entrega.
En la segunda parte de la entrega, se ha reasignado uno de los miembros del equipo B al equipo A y se han preparado dos nuevas tareas. El equipo A ha realizado la codificación del front-end y una pequeña parte del back-end, mientras que el equipo B se ha encargado de la arquitectura del sistema y el resto del back-end.

Arquitectura de desarrollo

La arquitectura de desarrollo del sistema sigue tal cual el patrón arquitectónico Modelo-Vista-Controlador (MVC). En este tipo de arquitectura se separan los datos, la lógica de negocio y la interfaz de usuario. Este patrón y esta separación hace más reutilizables los componentes de cada una de las partes sin tener que tocar otros. Por ejemplo, es posible modificar las vistas o la lógica de los datos sin necesidad de tocar el modelo (las entidades y la interacción con la base de datos).

Capa de acceso a datos

Esta capa se puede incluir o no dentro del modelo, dependiendo de lo detallada que sea la implementación del MVC. En este caso hemos creado un namespace aparte "database" que se ocupa del acceso a los datos.
Para el acceso a datos hemos implementado un patrón DAO (Data access object), que está representado en el diagrama de clases:

- ✓ **Interfaz DAO:** Define una interfaz para el acceso a datos. En el caso de que en determinado momento se quiera cambiar la base de datos MySQL por ficheros XML, por ejemplo, solo habría que añadir una nueva clase que implemente a

esta con los mismos nombres en los métodos. Nunca sería necesario hacer por tanto modificaciones en la vista o en la lógica (controladores), ya que aunque la parte de persistencia ha cambiado la interfaz y métodos a los que accede siempre son los mismos.

- ✓ **Clase abstracta SQLDAO:** Aquí se realiza el código común de todo el acceso a base de datos, es decir, el que compartirán todas las clases concretas que hereden de esta (BiddingDAO para subasta o ProductDAO para producto).
- ✓ **Clases DAO concretas:** Aquí se implementa código concreto para una clase concreta. En nuestro caso estas clases llaman solo al constructor de la clase abstracta, ya que se ha implementado de forma que en esta se genere código reutilizable para todas las clases de abajo.

Modelo

Los modelos se corresponden con las entidades que se muestran en el diagrama de clases. Los modelos representan a una entidad real (Usuario, Administrador, Producto…), y se ocupan de interactuar con la capa de acceso a datos.

El controlador se ocupa de interactuar con los modelos y estos contienen métodos para realizar operaciones con la base de datos, ya sea modificándola u obteniendo datos de esta.

Entre las características del modelo en el diagrama de clases es importante hablar de otras dos clases (DAOFactory y Model).

DAOFactory es una clase del paquete de acceso a datos que sirve para crear DAOs en función de lo que se necesite. Un ejemplo sería cuando se añade una nueva subasta, para añadirla a la base de datos hay que crear un nuevo DAO y para esto se puede pasar como parámetro el nombre de la propia entidad "subasta". Consideremos que se añade una nueva subasta (clase Bidding), pasando al DAOFactory el nombre de esta clase este creará, en base al nombre de esta entidad un SQLBiddingDAO, si en vez de eso se pasase "Product" se crearía un SQLProductDAO. Este fenómeno se llama Reflexión (modifica en tiempo de ejecución el comportamiento del programa en base al nombre de la entidad, y el patrón de creación es una factoría, que crea nuevos objetos dependiendo de un parámetro que se le pase.

Model es una clase abstracta con las operaciones comunes de todos los modelos, es decir, el contacto con la capa de acceso a datos para almacenar, borrar y validar.

Controlador

Dentro del paquete de controladores se incluye la lógica de negocio de cada clase concreta (un controlador por entidad). Los controladores se encargan de abstraer las vistas de los modelos, gestionando su interacción con ellos y haciéndolos así más reutilizables (el cambio en un modelo no afecta a las vistas y viceversa). Los controladores disponen de una clase abstracta común para compartir algunos métodos para todos los controladores.

Dentro del paquete de controladores se contiene también una clase Router que se encarga, según la petición, de elegir el controlador concreto que se va a usar, crearlo y llamar a la acción (método) correspondiente.

Vista

La clase View se asocia con las distintas plantillas y carga los archivos correspondientes a cada vista según la acción definida por el controlador asociado a dicha vista.

Normas de estandarización

Base de datos

Nomenclatura

Todas las tablas deberán tener un nombre escrito completamente en mayúsculas, y siempre en singular. No se admitirá, por ejemplo, una tabla llamada "Ventas", debería ser "VENTA".

Las columnas de cada una de las tablas seguirán una nomenclatura camel case, con la primera letra en mayúscula. Por ejemplo, puede existir una columna llamada "idVenta", pero no se admitirá "id_venta" o "idventa".

Aspectos a considerar

Se prefiere la existencia de una única clave primaria en cada tabla. Es decir, que la clave primaria está conformada por tan solo una columna. No es obligatorio seguir esta norma, pero es recomendable puesto que facilitará la posterior codificación del sistema en PHP.

Se deberá hacer uso de claves secundarias siempre que sea necesario, y éstas deberán tener especificado el modo de borrado/actualización necesario cuando se modifique la tabla original (CASCADE, SET NULL…).

Se preferirá el uso de enums frente al uso de un integer para denotar un tipo. Por ejemplo, para marcar que un usuario es administrador o no, podría utilizarse un valor entero donde 0 se corresponde a usuario y 1 a administrador, en este caso se preferirá la existencia de un enum donde se definan los valores 'administrador' y 'usuario'.

Código PHP

Indentación

Todo el código PHP debe estar correctamente indentado. Una indentación debe consistir de 4 espacios. El uso de tabuladores no está permitido.

Se recomienda la configuración del editor de texto utilizado acorde a esta norma, para que cuando se pulse la tecla tabulador se traduzca automáticamente a 4 espacios, y para que las indentaciones automáticas también se realicen con 4 espacios.

Máxima longitud de línea

La longitud máxima para cualquier línea de código será de 120 caracteres. La longitud recomendada es de 80 caracteres. Se deberá intentar mantener las líneas de código por

debajo del límite recomendado, y sólo bajo circunstancias excepcionales se permitirá el sobrepaso del límite máximo.

Terminación de línea

La terminación de línea seguirá las convenciones Unix. Esto es, las líneas deben terminar única y exclusivamente con un único carácter de nueva línea (linefeed, LF). No se permitirá la terminación de línea de las convenciones de Apple (carriage return, CR) ni las de Windows (carriage return y linefeed, CRLF).

Se recomienda configurar el editor utilizado para que se adapte a esta norma.

Delimitación de bloques (llaves)

Las llaves de apertura y cierre de bloques deberán escribirse en sus propias líneas siempre que se correspondan con bloques de clases, métodos o funciones. En cualquier otro caso las llaves de apertura deberán estar en la misma línea que la expresión que abre el bloque (ifs, fors, whiles…).

Ejemplo:

```
class MiClase
{
  public function miFuncion($variable)
  {
     if (isset($variable)) {
        print $variable;
     }
  }
}
```

Demarcación

Todo código PHP debe estar delimitado siempre por la forma completa de los tags estándar de PHP:

```
1. <?php
2.
3. ?>
```

Los tags reducidos no están permitidos bajo ningún concepto. Debe tenerse en cuenta que, en un fichero que cuente solamente con código PHP, tras el tag de cierre ?> no debe incluirse **nunca** y bajo ningún concepto una nueva línea en blanco.

Cadenas de texto

Toda cadena de texto en PHP debe estar delimitada siempre por comillas dobles, el uso de comillas simples no será aceptado.

```
1. $a = “Ejemplo de String”;
```

La sustitución de variables dentro de una cadena se admitirá siempre bajo estas formas:

```
1. $saludo = “Hola $nombre, bienvenido”;
2.
3. $saludo = “Hola {$nombre}, bienvenido”;
```

Y nunca, por consistencia, bajo la forma alternativa:

```
1. $saludo = “Hola ${nombre}, bienvenido”;
```

La concatenación de cadenas debe realizarse siempre con el operador “.”, que debe estar siempre precedido y seguido de un espacio.

```
1. $asignatura = “Interfaces” . “ de “ . “Usuario”;
```

Clases

Las clases seguirán una nomenclatura camel case con la primera letra en mayúsculas. Los métodos y atributos seguirán también una nomenclatura camel case, pero con la primera letra en minúsculas.

```
class NombreClase
{
        private $atributoUno;

        public funciton nombreFuncion()
        {
          ...
        }
}
```

Todo método y atributo dentro de una clase debe declarar su visibilidad, siendo ésta una de: public, protected o private. La declaración de atributos como var, sin especificar su visibilidad, no estará permitida.

La estructura de directorios, ficheros y namespaces seguirá la normativa PSR-0 para la correcta carga automática de las mismas. Es decir, cada clase deberá estar definida en un fichero con el mismo nombre de la clase (respetando mayúsculas), y siempre que se defina dentro de un namespace, dicho fichero .php deberá almacenarse dentro de un directorio con el nombre del namespace.

Por ejemplo, si definimos la clase MiClase dentro del namespace miespacio, definido a su vez dentro del namespace espacios, el fichero .php que define la clase deberá encontrarse, partiendo de la raíz del directorio de código, en:

/espacios/miespacio/MiClase.php

Sistema de comunicación y control de cambios

Diagrama de casos de uso

Se ha trabajado sobre la ET1 considerando ciertas modificaciones desde el inicio en el prototipado falso. Se ha modificado un pequeño detalle en el diagrama de casos de uso

en relación a las tareas del usuario, que consideramos que no debe modificar ni eliminar subastas. En caso de que alguien puje no tendría lógica que sea posible que el usuario pueda eliminar el producto de la subasta. Quizás se podría considerar también que se eliminase si no hay pujas, pero no hemos entrado a ese nivel de detalle.

Prototipado falso

El grupo A se ha ocupado de retocar el prototipado según iban avanzando las vistas, aunque la versión final se ha dejado para la última semana, ya que según iba avanzando el sistema funcional se iban encontrando nuevos errores de prototipado, por lo cual no tenía sentido realizar excesivos cambios en ese punto y tampoco en estado y transiciones. El prototipado se ha modificado en varias ocasiones pero no se ha realizado una versión final de todo hasta el último día, debido a lo comentado.

Considerando estos problemas con el prototipado, se ha realizado la nueva versión de estados y transiciones considerando el último prototipado solo una vez, ya que no consideramos productivo en este caso gastar recursos en hacer modificaciones continuas de ese punto. Hemos ido modificando solo pequeños detalles, dándole así un poco más ágil al desarrollo.

ChangeLog

- Análisis de requisitos
    - Eliminada la posibilidad de modificación de pujas.
    - Eliminada la posibilidad de que el usuario elimine sus pujas. Solo el administrador debe poder eliminarlas. (*no se especifica eliminación de subastas)
- Diagrama de casos de uso.
    - Se ha eliminado la posibilidad de modificar pujas.
    - El usuario ya no puede eliminar pujas, en cambio el administrador sí.
- Descripción de casos de uso
    - Eliminación de “Modificar Pujas”
    - Ahora solo el administrador puede eliminar pujas.
    - Modificada posibilidad de cambiar porcentajes al administrador desde vista usuario, ahora solo es posible realizarla desde la vista administrador.
- Diagrama entidad-relación
    - Se creó una entidad Calificación relacionada con los productos y los usuarios con los atributos idCalificacion, puntuacion y comentario.
    - Se eliminaron los atributos cantidad y tipo de la entidad Producto.
    - Se añadió el atributo estado a la entidad Producto.
    - Se cambió la cardinalidad de la relación Producto-Subasta a (0,1).
    - Se eliminaron los atributos nombre, tiempoLimite, descripción y fecha de la entidad Subasta.
    - La entidad Puja se transformó en entidad débil para facilitar la implementación.
    - Se eliminaron los atributos fecha, hora y secuencia de la entidad Puja y se añadieron idPuja y fechaPuja.
    - Se añadió la relación Puja - Pago (1,N)

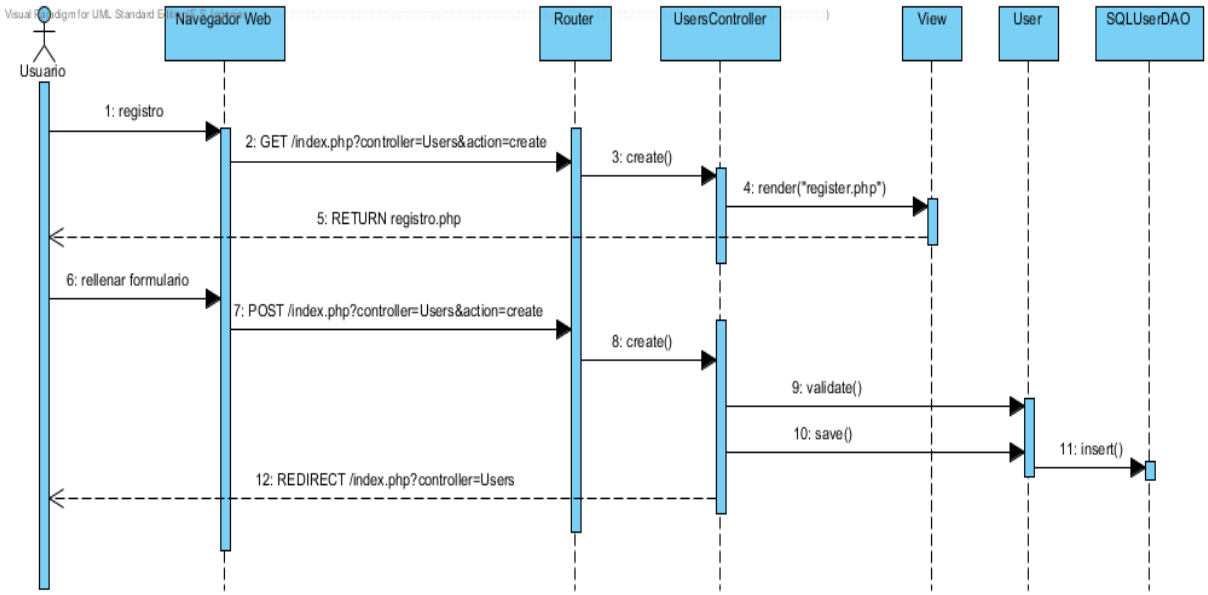
    - Se añadieron los atributos: cuentaPaypal, numTarjeta, metodoPago y comision a la entidad Pago
    - Se creó la entidad Compra relacionada con Venta, Pago y Usuario con los atributos idCompra, fechaCompra y cantidad.
    - Se cambio la cardinalidad de la relación Producto Venta a (0,1).
    - Se añadió el atributo stock a la entidad Venta.
    - Se eliminaron los atributos: cuentaPaypal, cuentaBancaria, fechaNacimiento y movil de la entidad Producto.
    - El atributo telefono de la entidad Producto pasó a ser simple.
    - Se creó una relación Usuario - Puja (0,N)
- Prototipado falso
    - El prototipado falso se ha vuelto a hacer de cero al igual que el diagrama de estados y el diagrama de transiciones.
- Diagrama de estados.
- Diagrama de transiciones.

Sistema de comunicación entre los miembros del grupo

Para la comunicación entre los dos equipos se ha usado principalmente un grupo de Whatsapp, que ha servido especialmente para formalizar reuniones continuas así como acordar fechas y horas para realizar partes del trabajo entre los miembros. Para noticias o información larga y más detallada se ha hecho uso del correo electrónico.

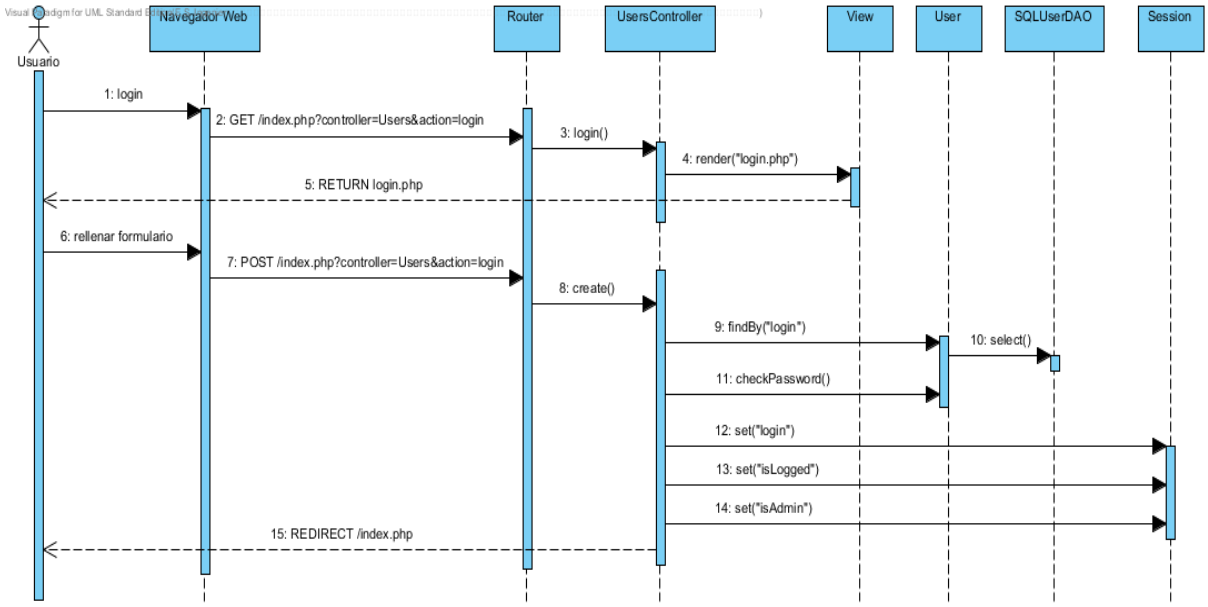
Diagramas de secuencia detallados

Registrar usuario



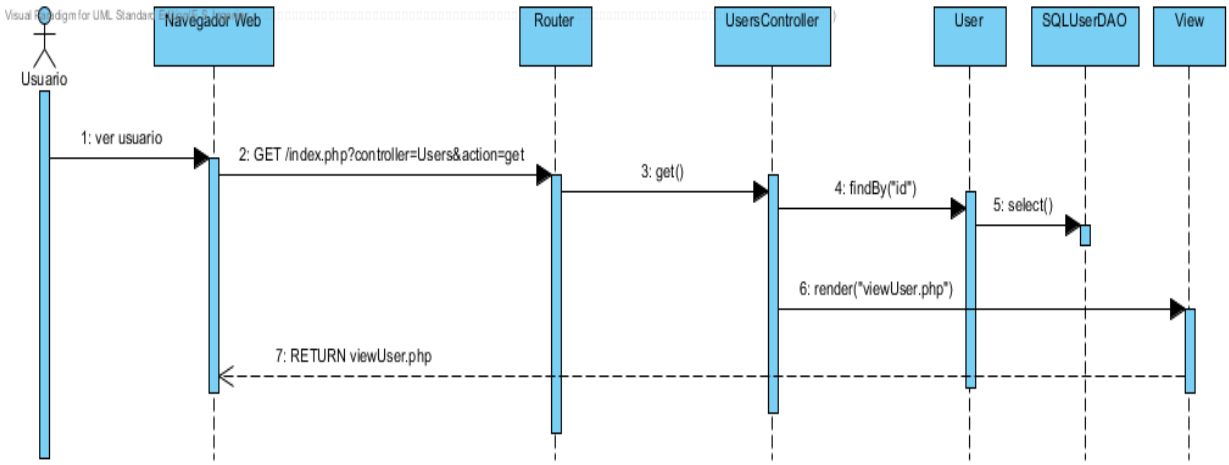
Pantalla número 2 del prototipado falso

Login usuario



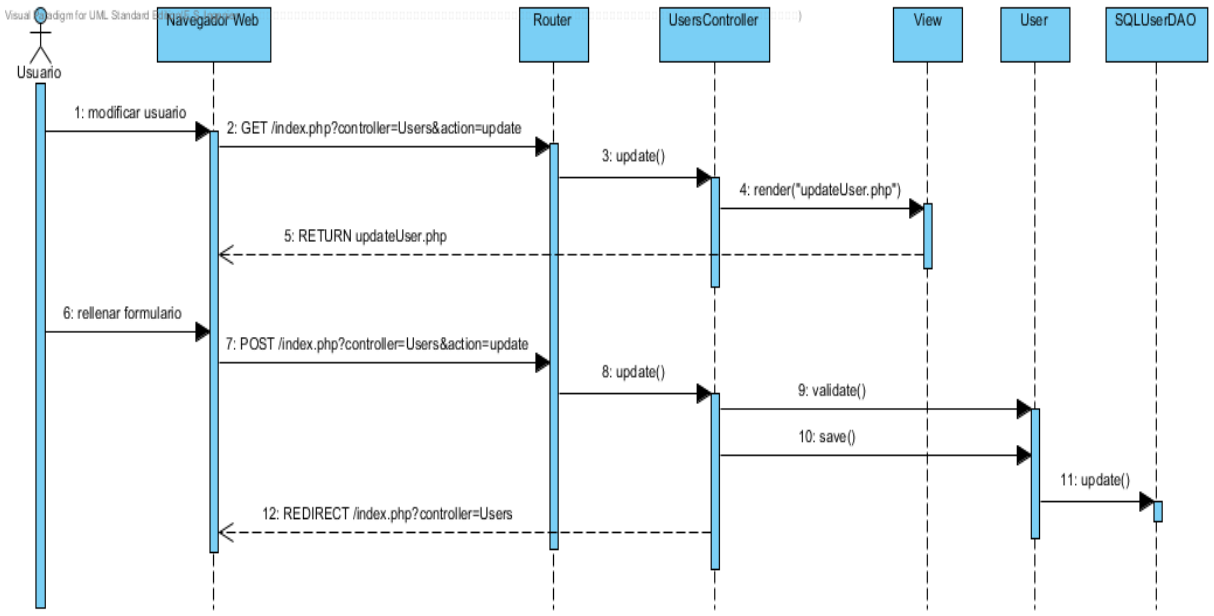
Pantalla número 3 del prototipado falso

Consultar usuario



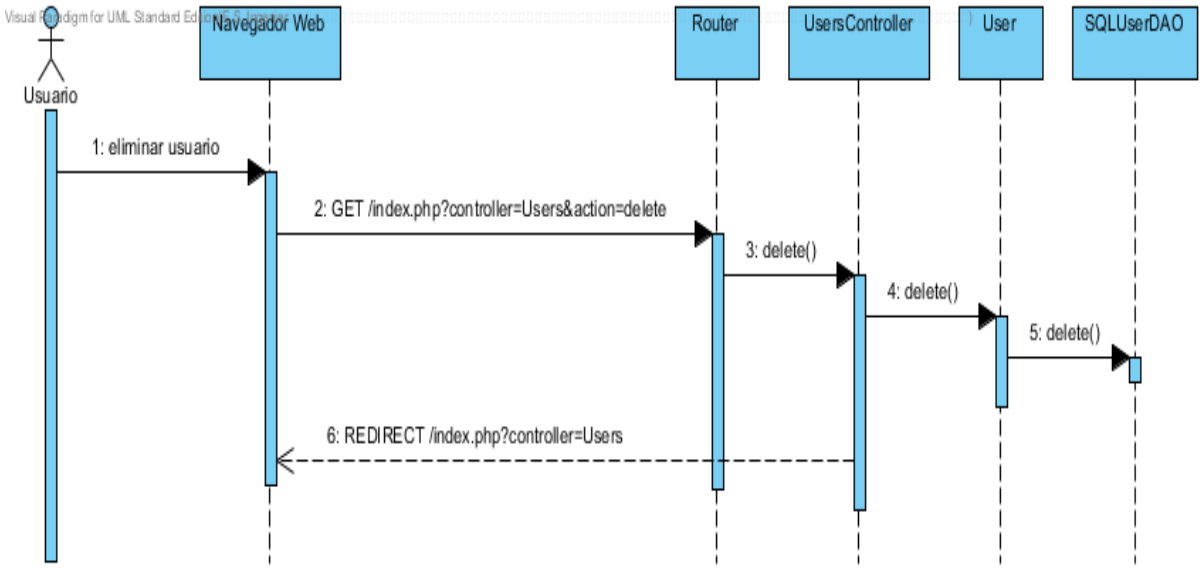
Pantalla número 6 del prototipado falso

Modificar usuario

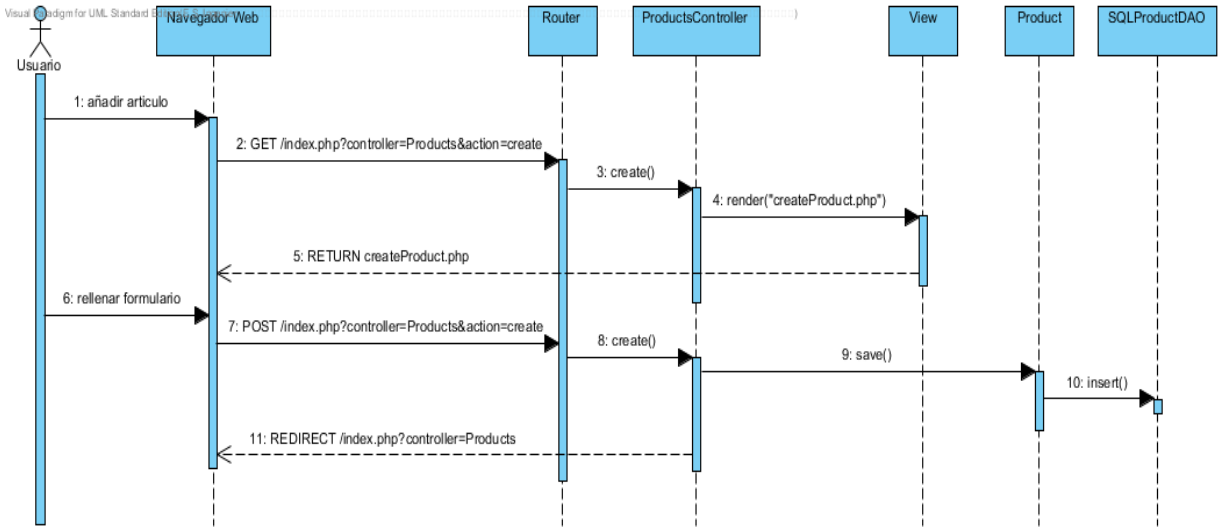


Pantalla número 13 del prototipado falso

Eliminar usuario

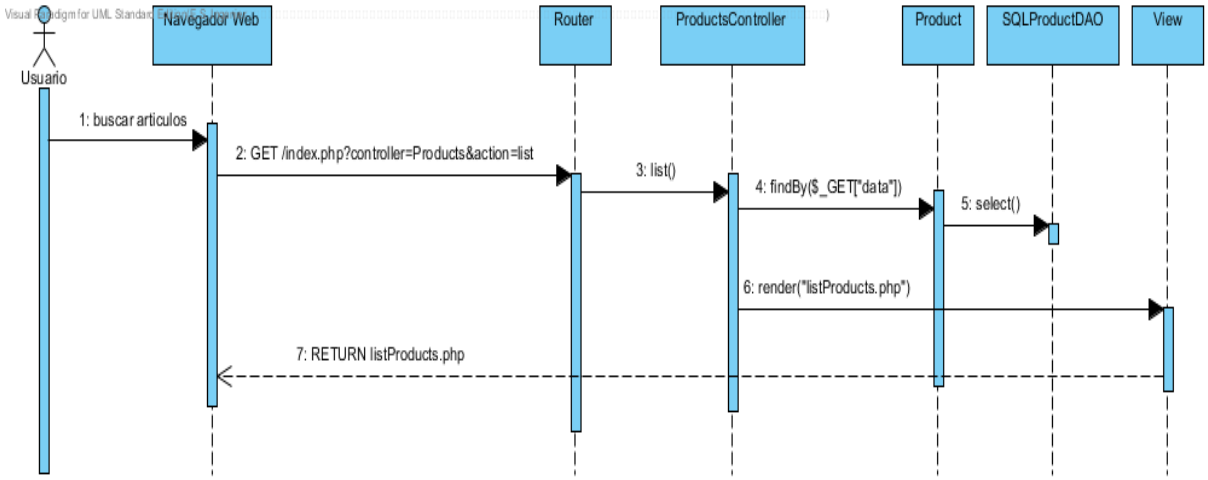


Añadir artículo

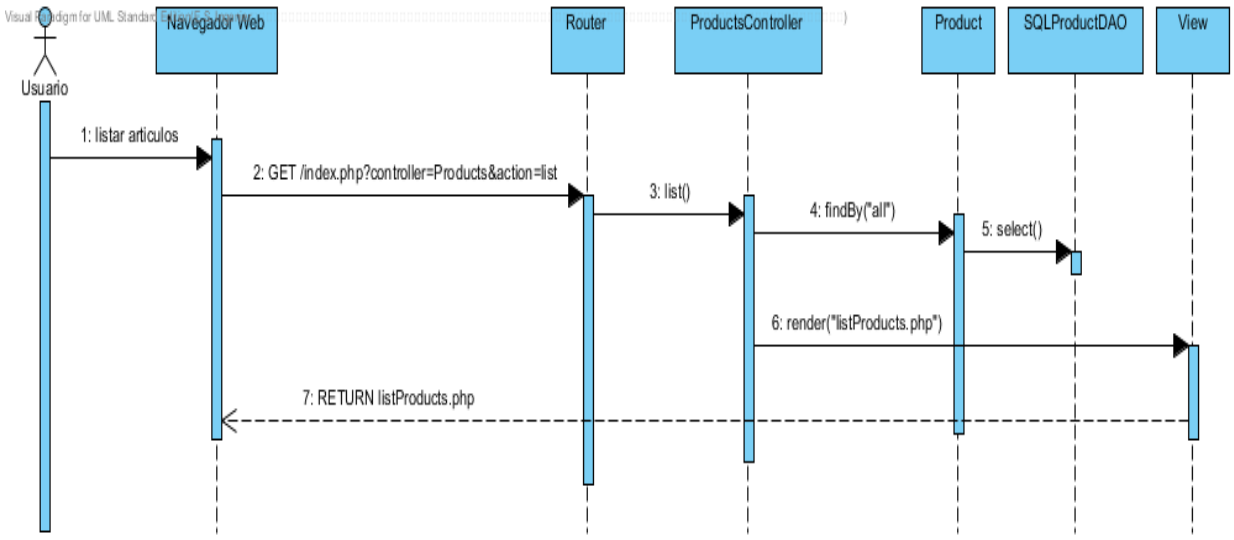


Pantalla número 14 del prototipado falso

Buscar artículos

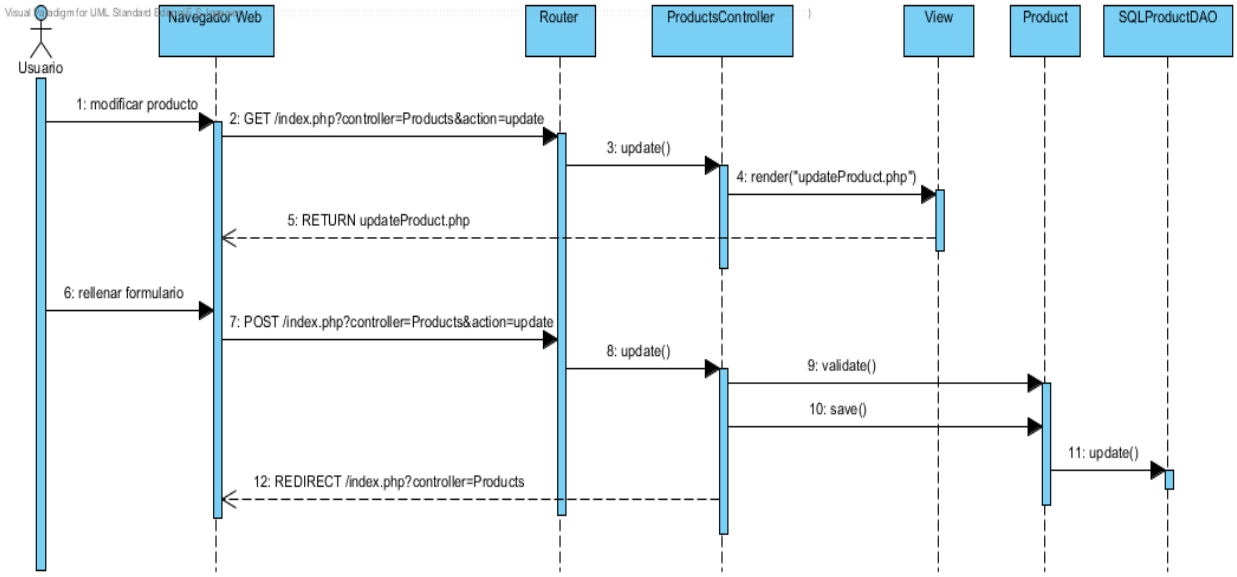


Listar artículos



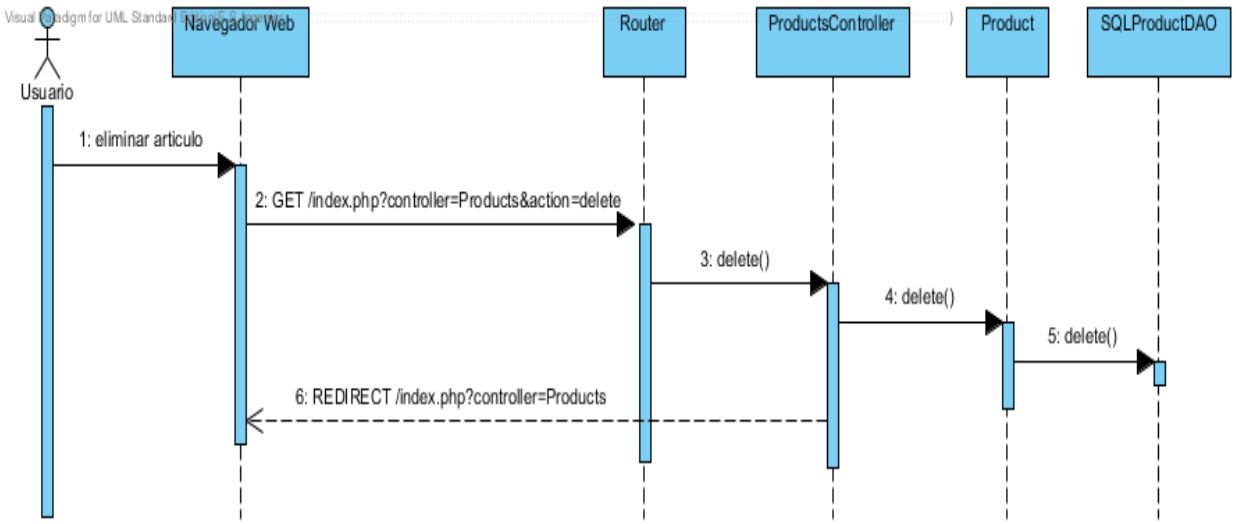
Pantalla número 5 del prototipado falso

Modificar artículo

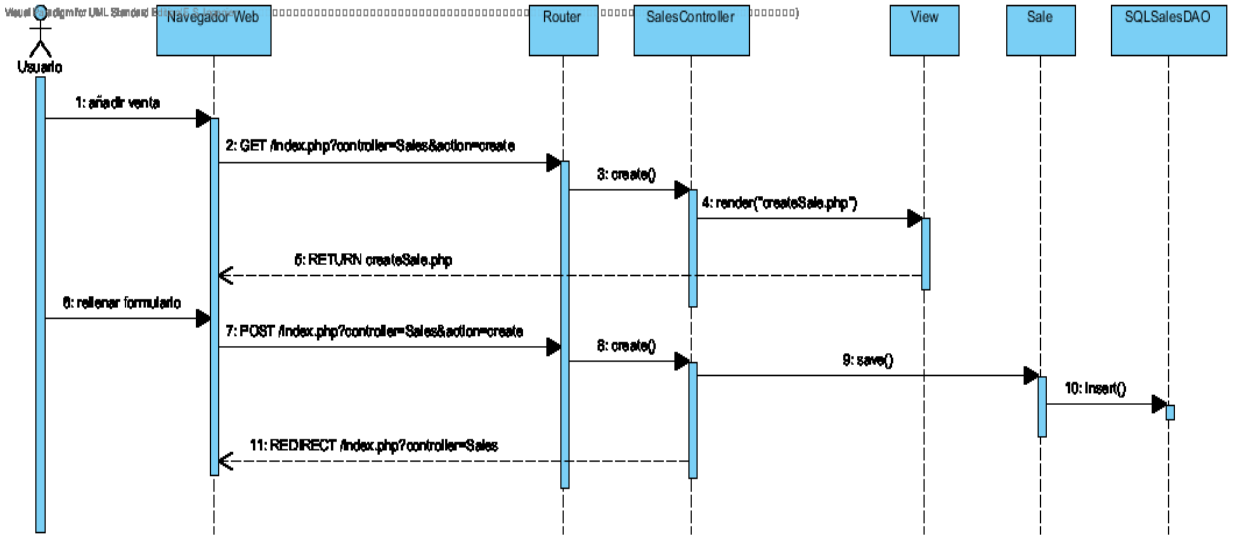


Pantalla número 8 del prototipado falso

Eliminar artículo

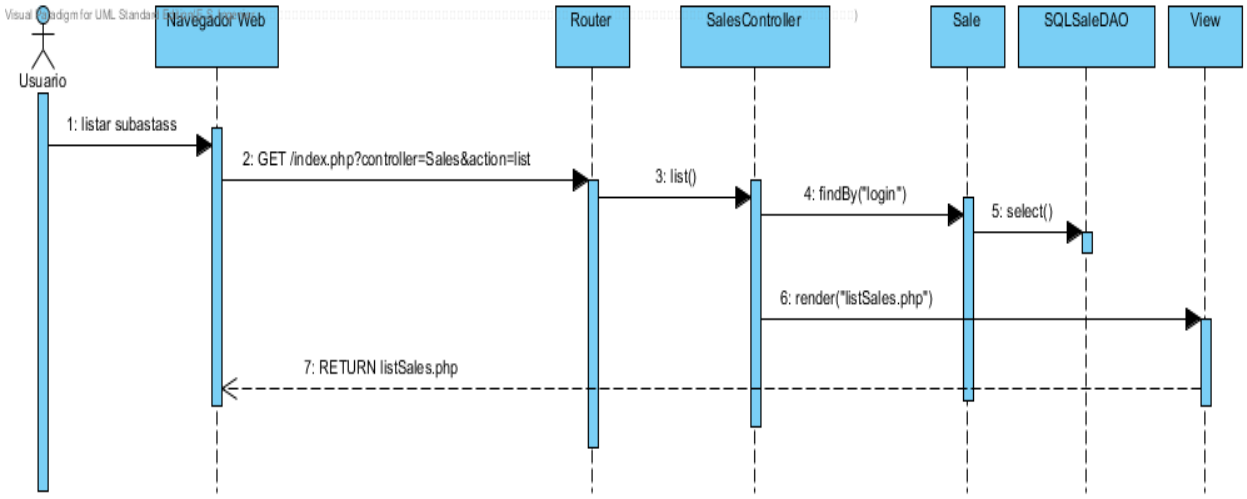


Añadir venta



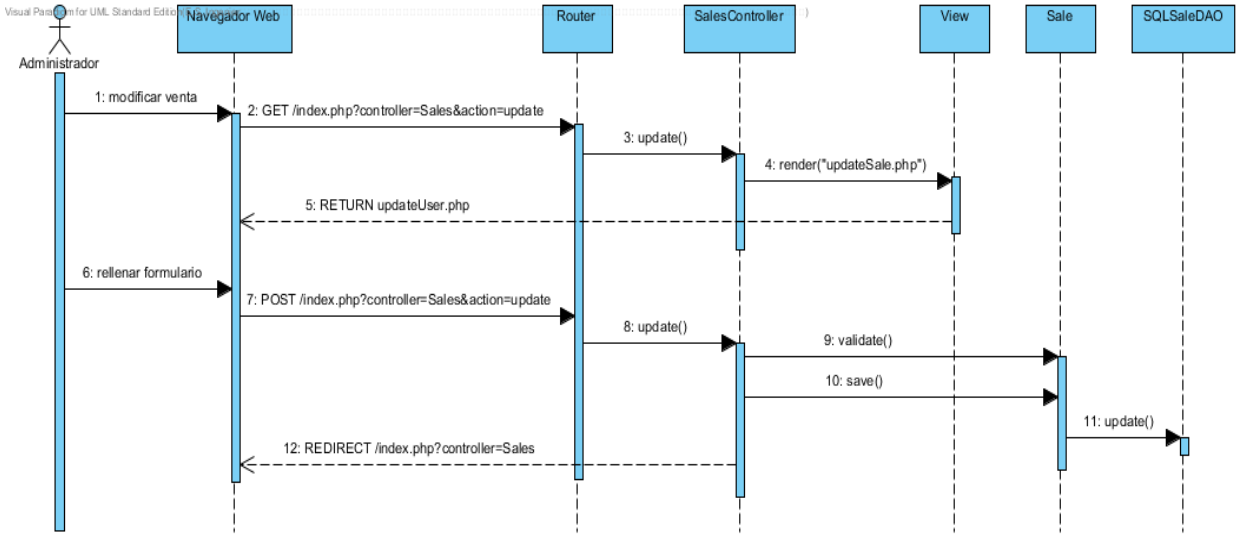
Pantalla número 10 del prototipado falso

Listar ventas



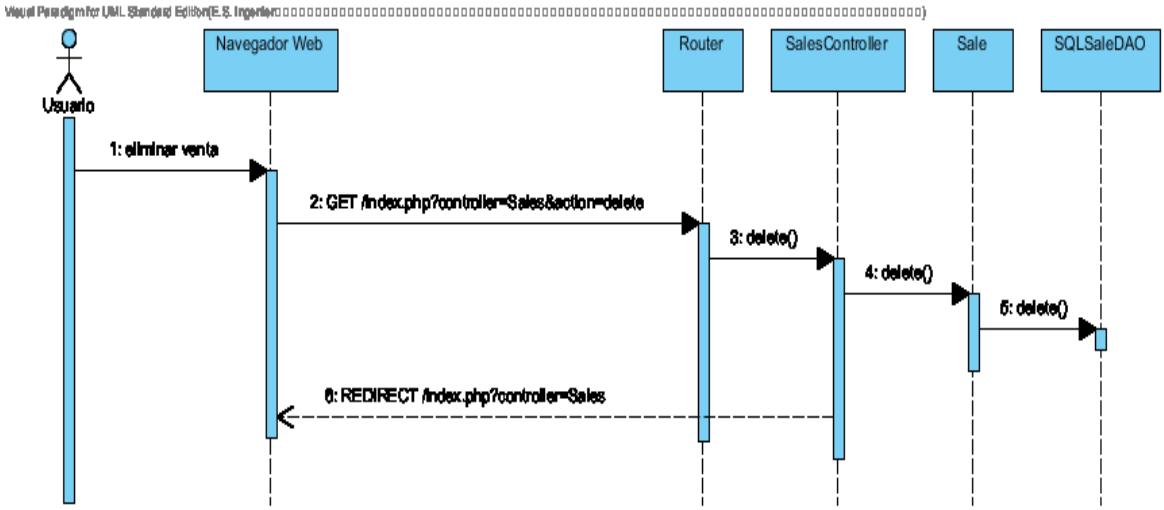
Pantalla número 26 del prototipado falso

Modificar venta

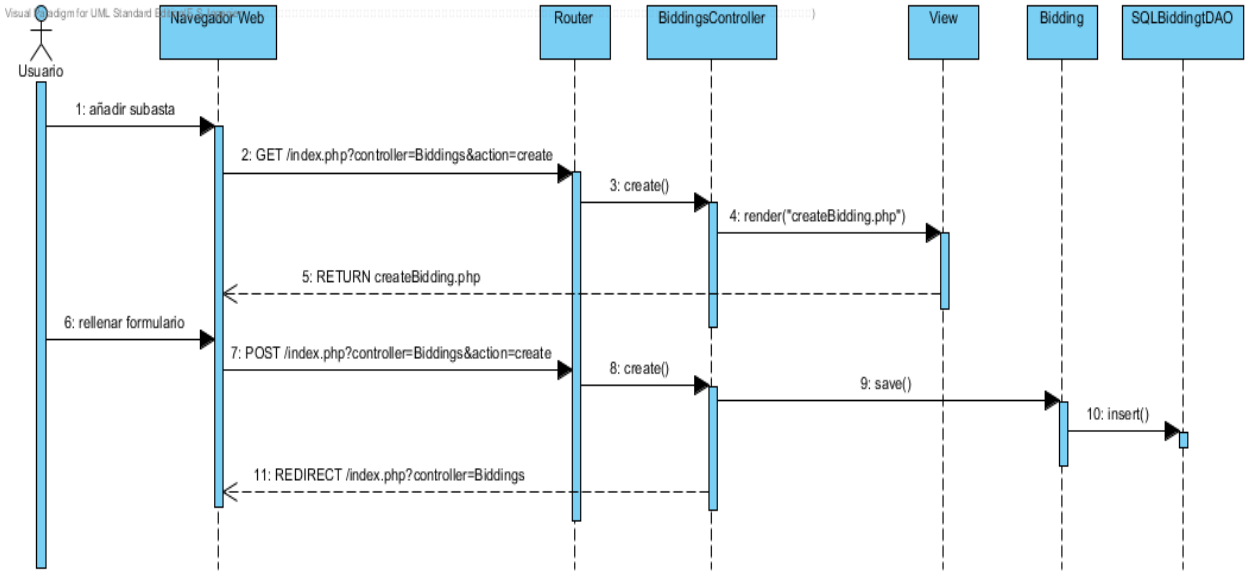


Pantalla número 24 del prototipado falso

Eliminar venta

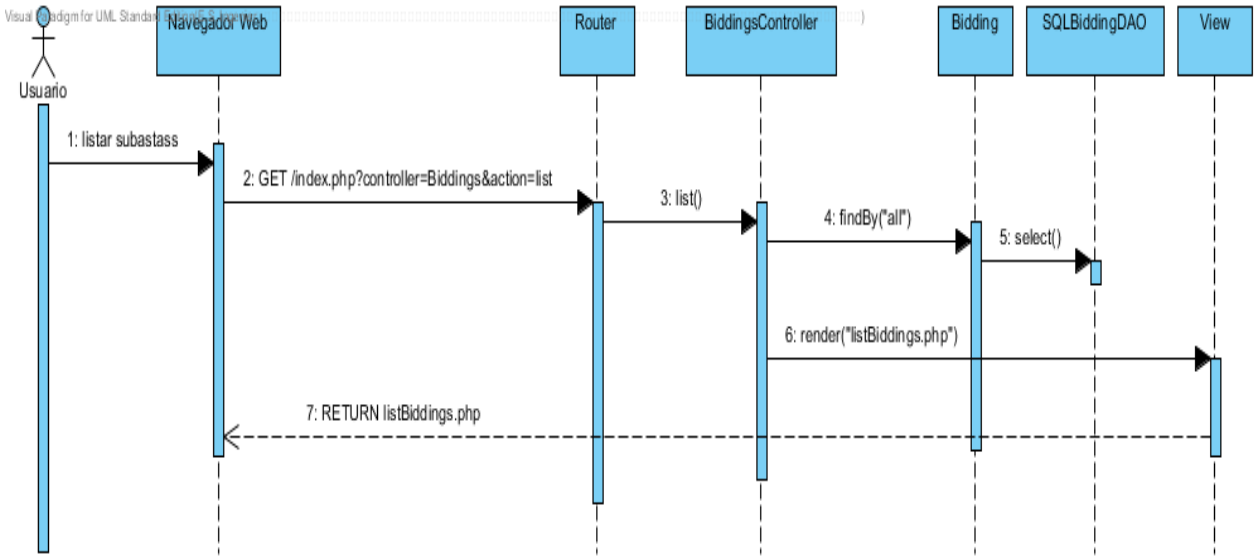


Añadir subasta



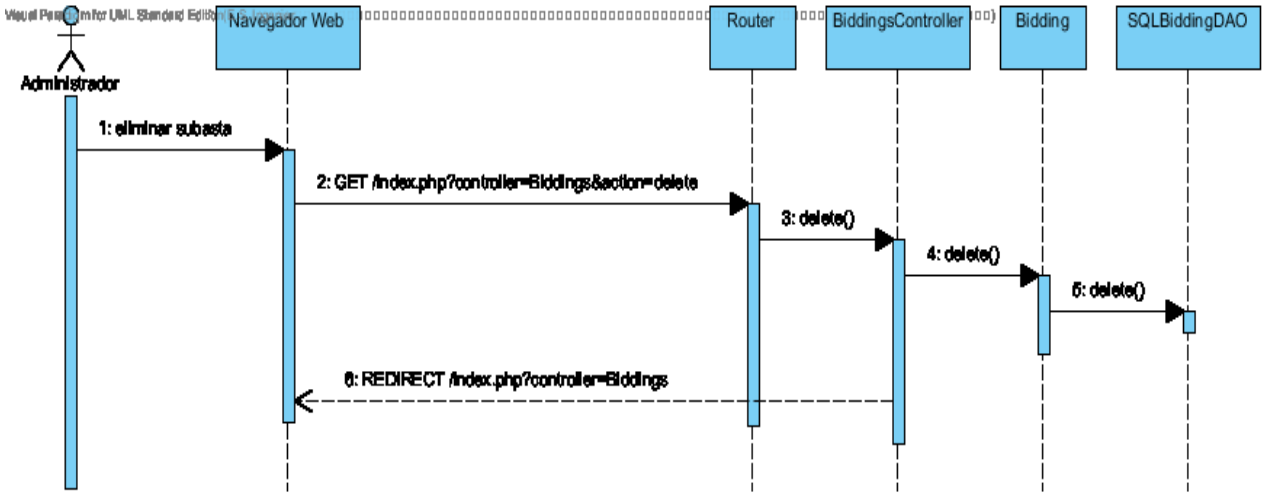
Pantalla número 12 del prototipado falso

Listar subasta

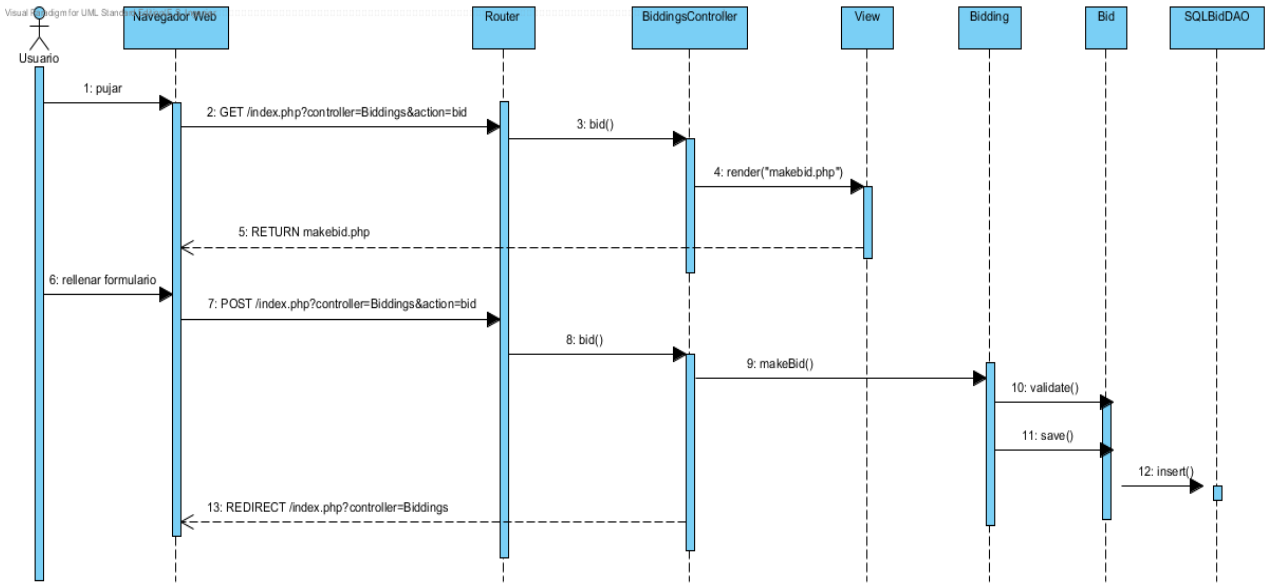


Pantalla número 27 del prototipado falso

Eliminar subasta

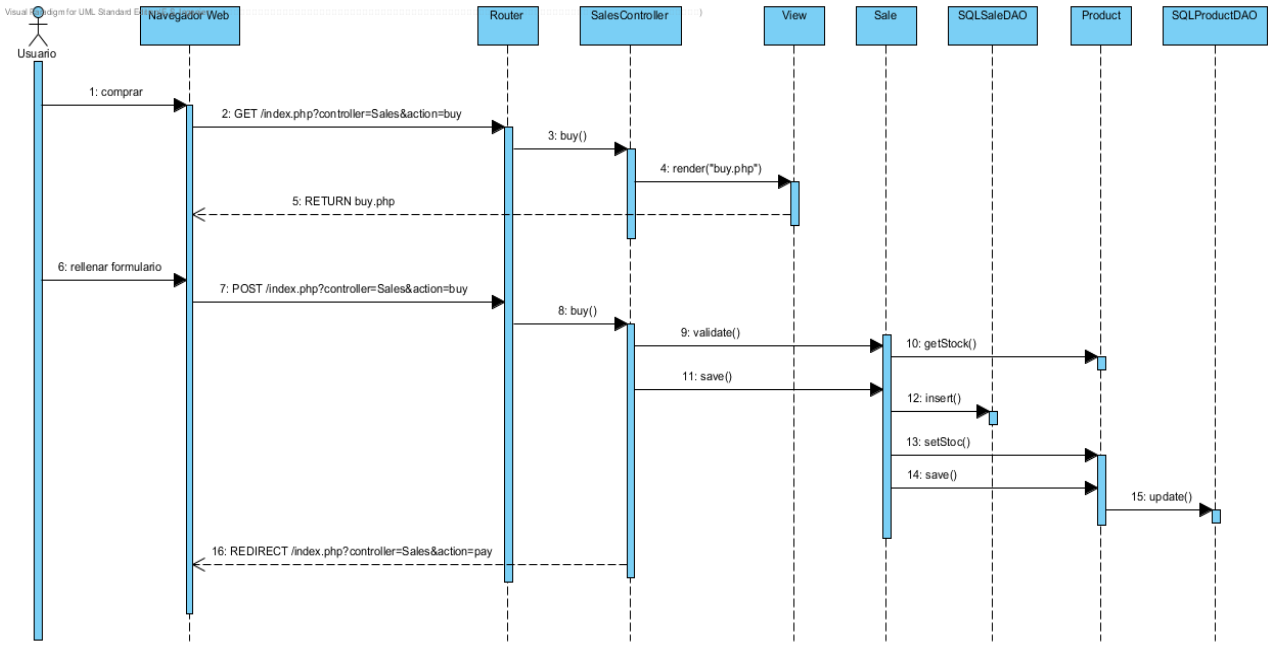


Pujar



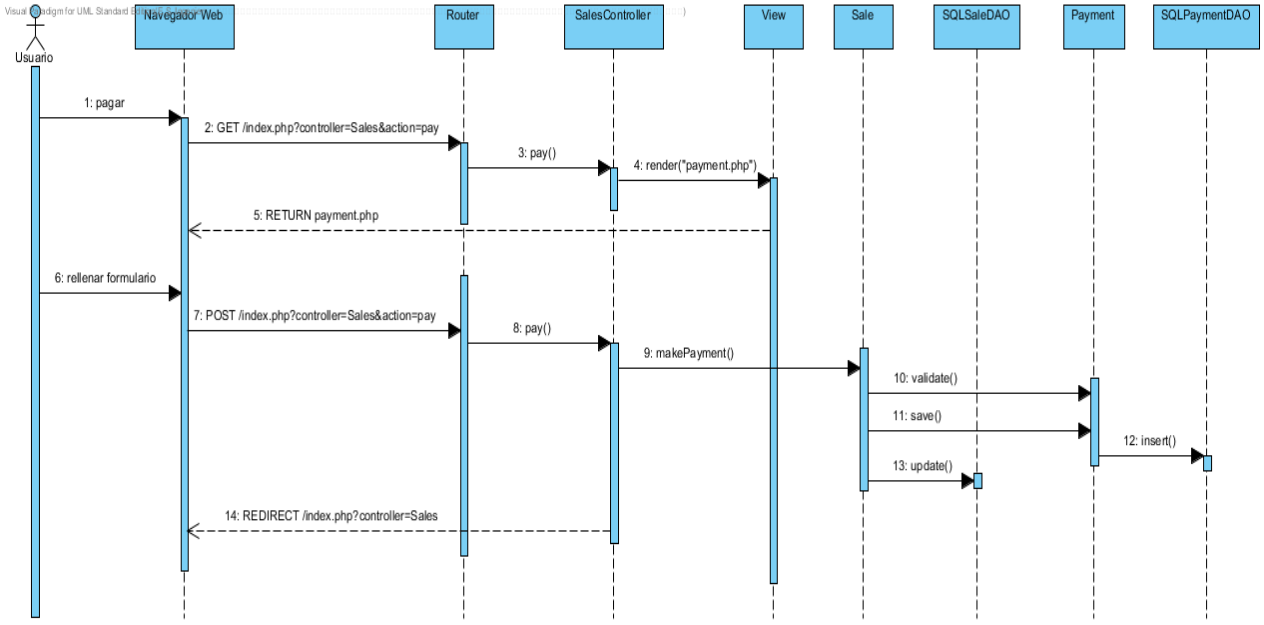
Pantalla número 20 del prototipado falso

Comprar



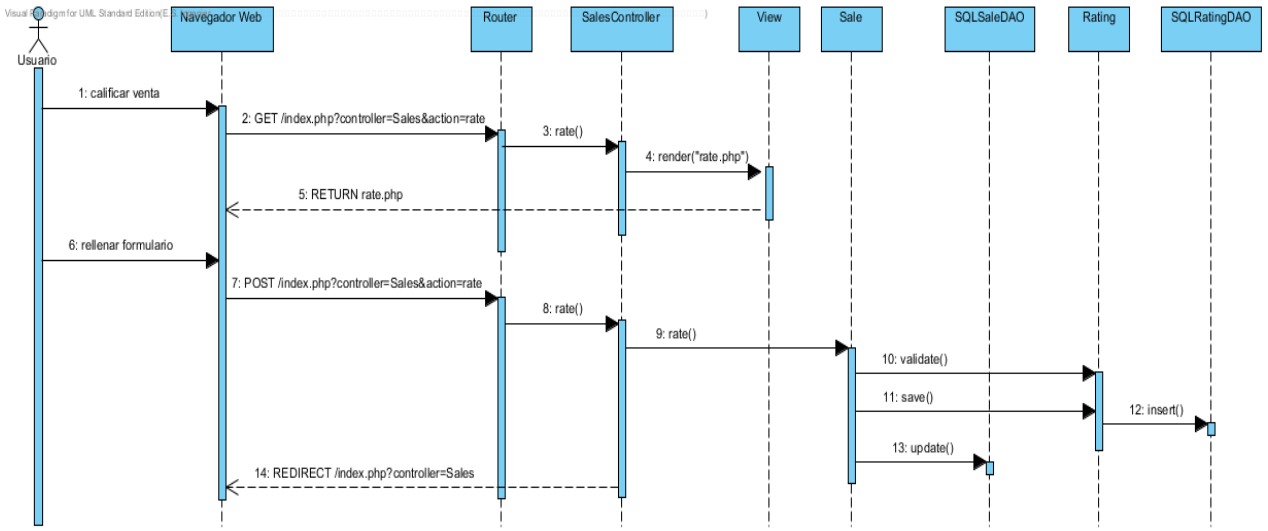
Pantalla número 18 del prototipado falso

Realizar pago



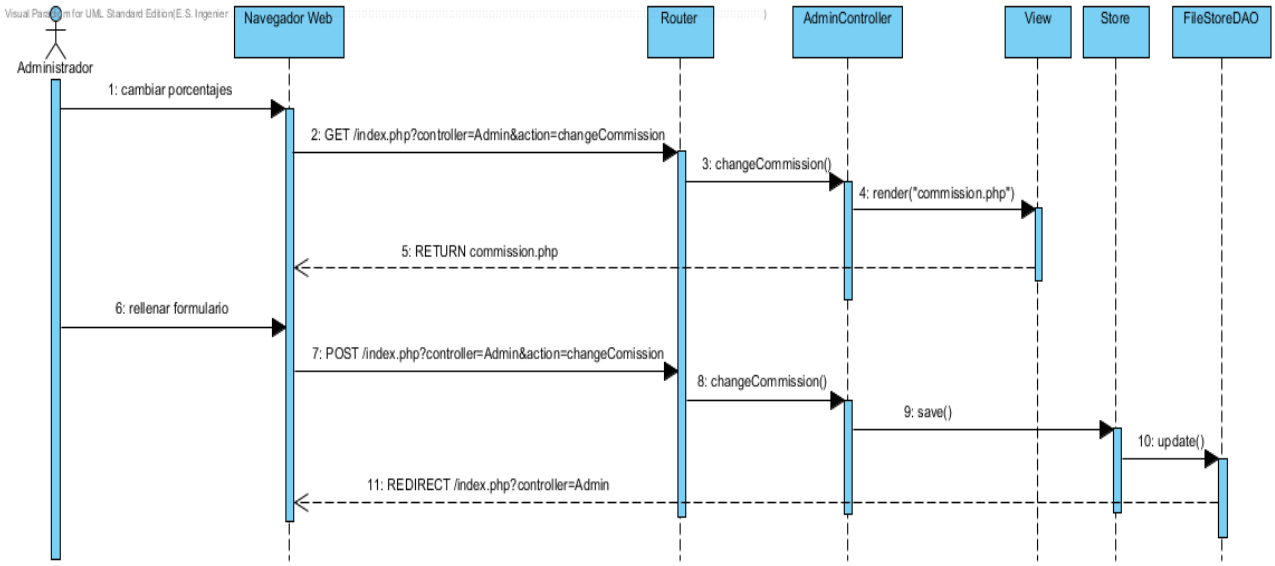
Pantalla número 34 del prototipado falso

Calificar venta



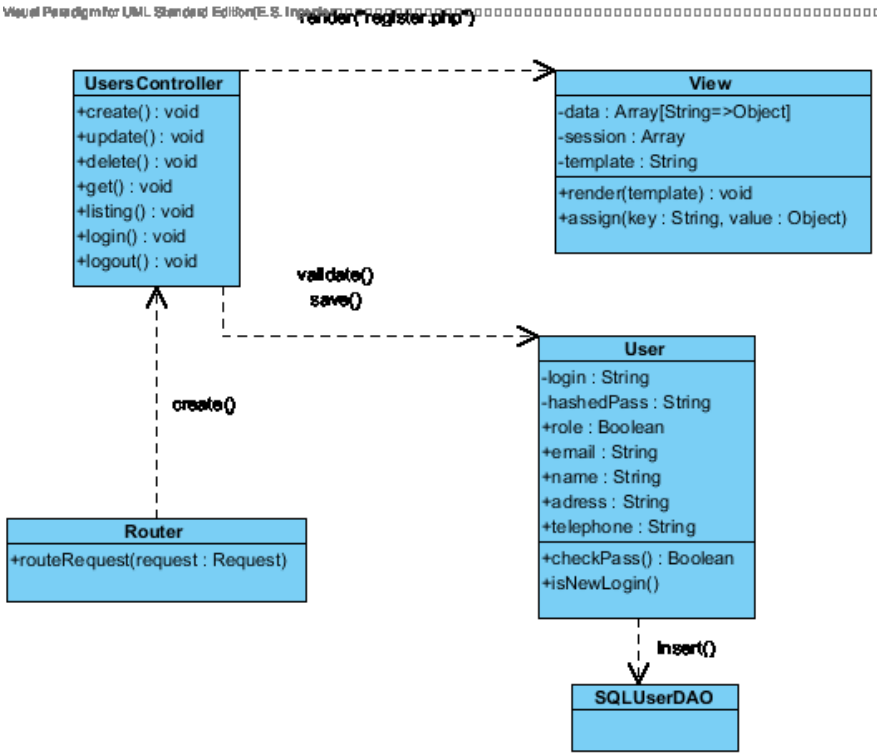
Pantalla número 9 del prototipado falso

Cambiar porcentajes

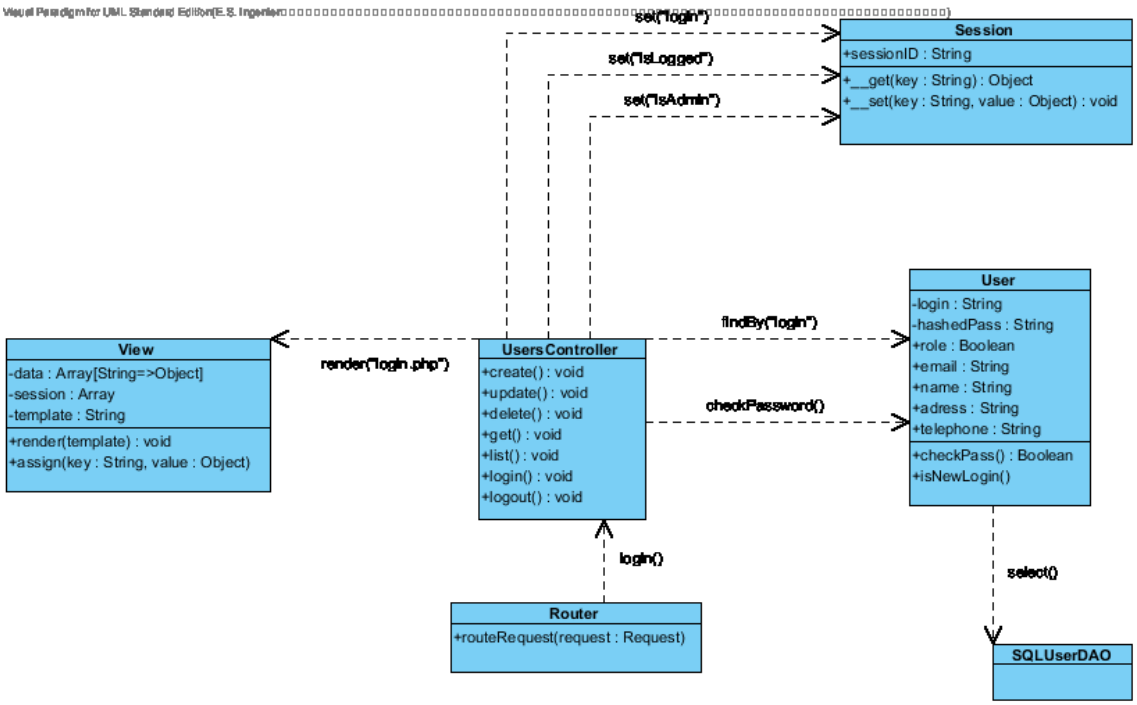


Diagramas de clases parciales

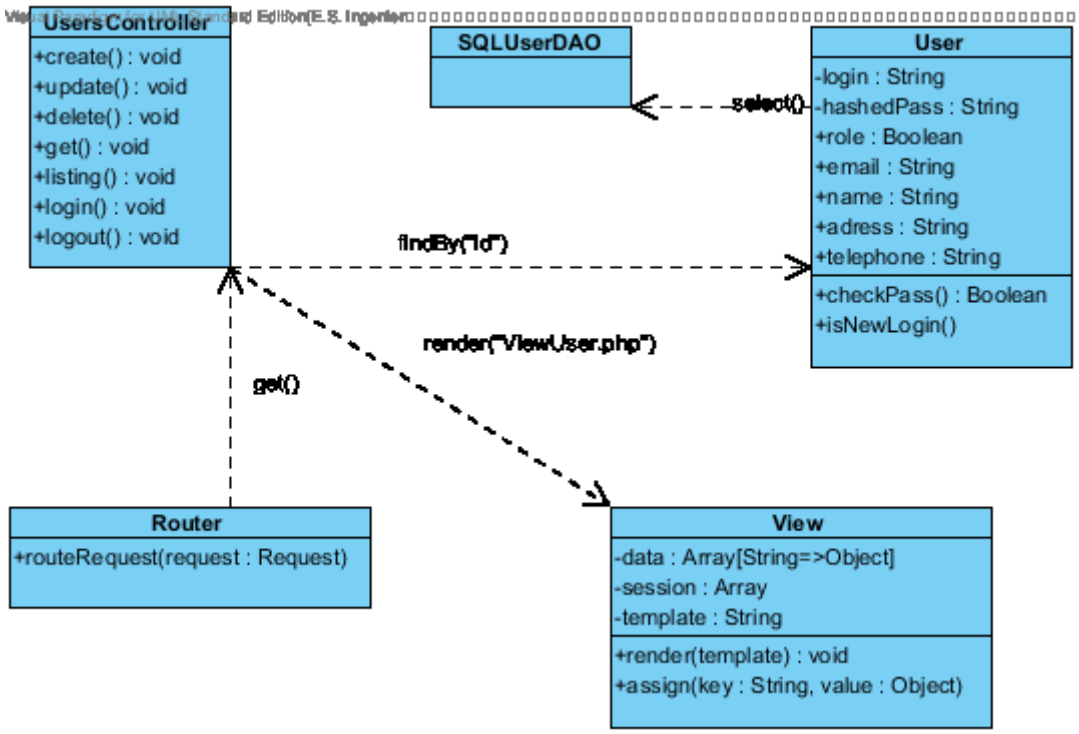
Registrar usuario



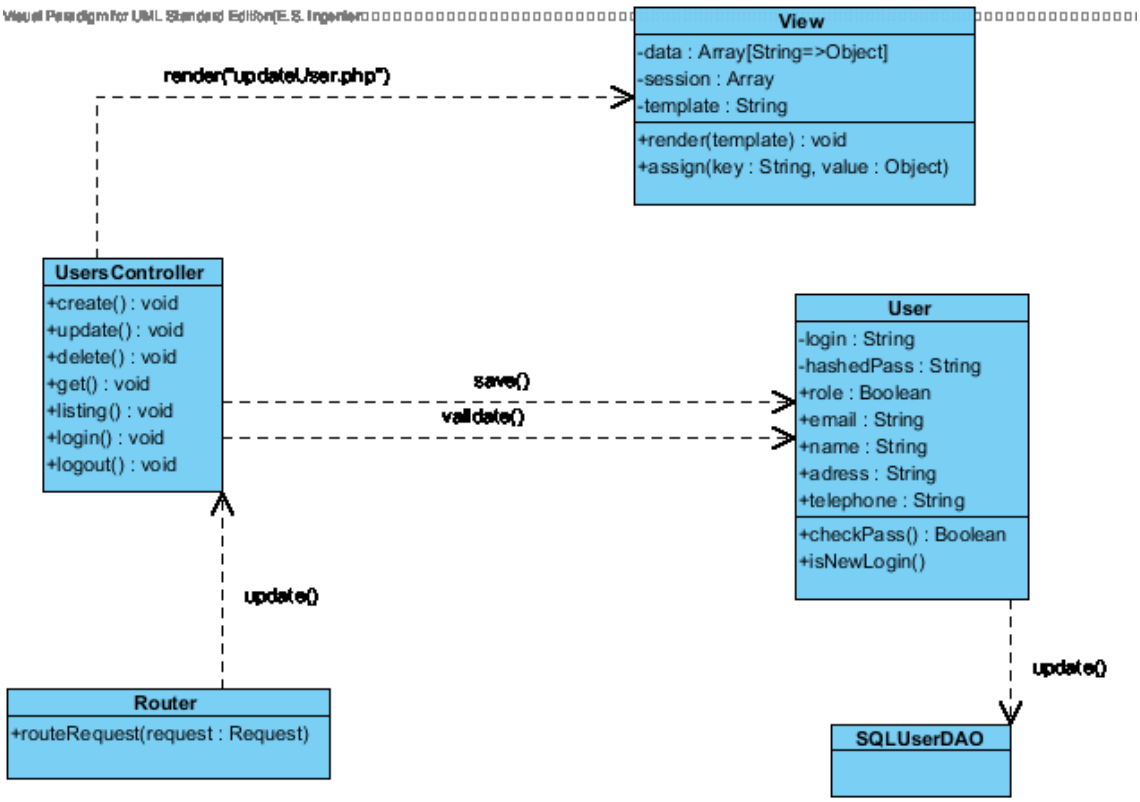
Login usuario



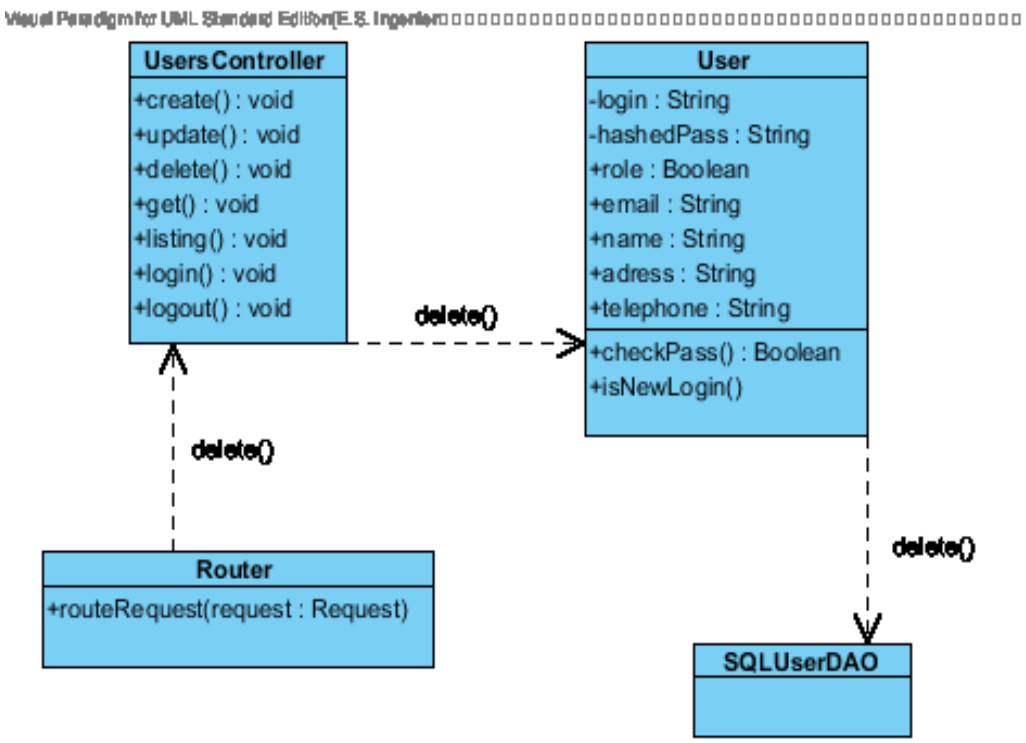
Consultar usuario



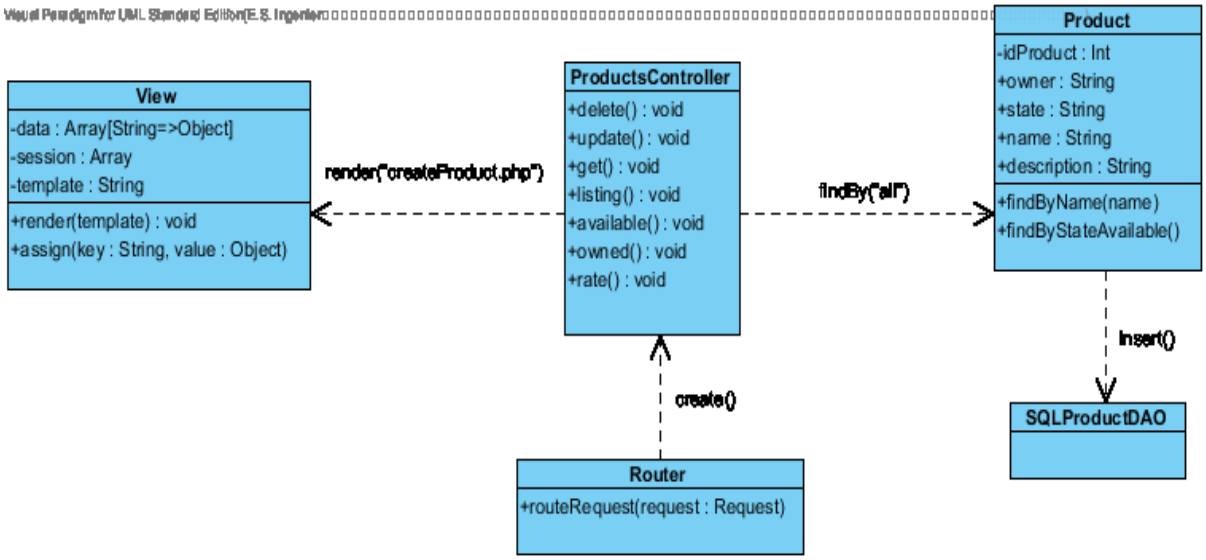
Modificar usuario



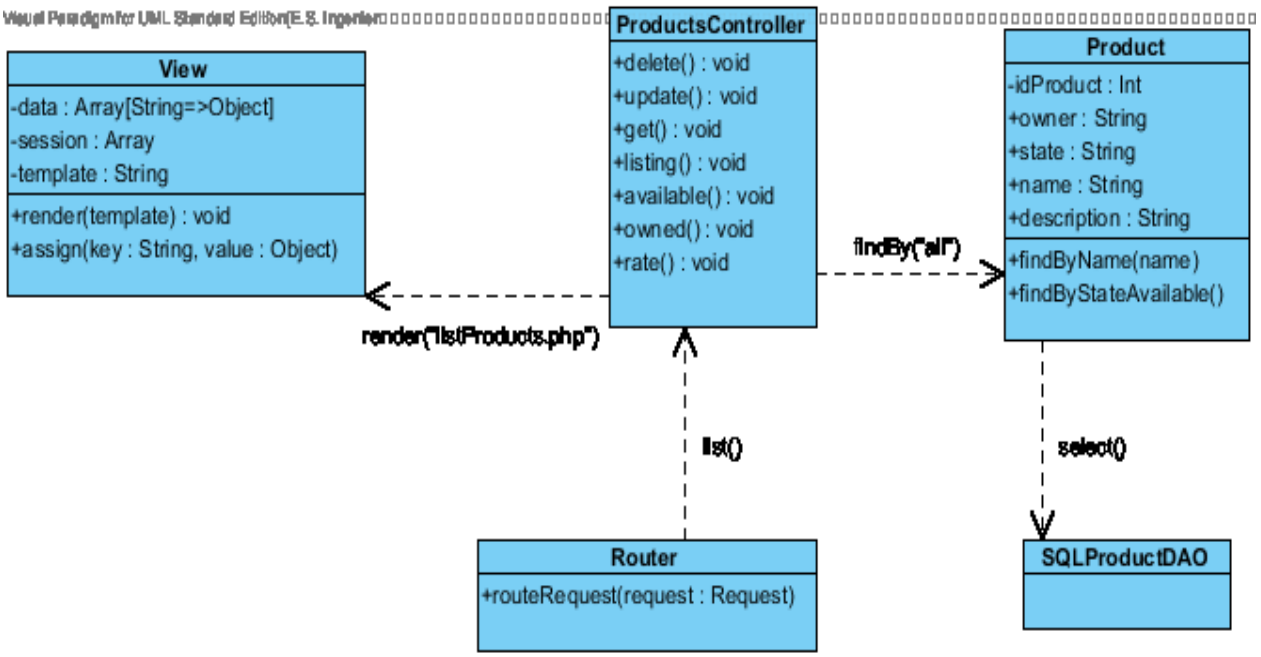
Eliminar usuario



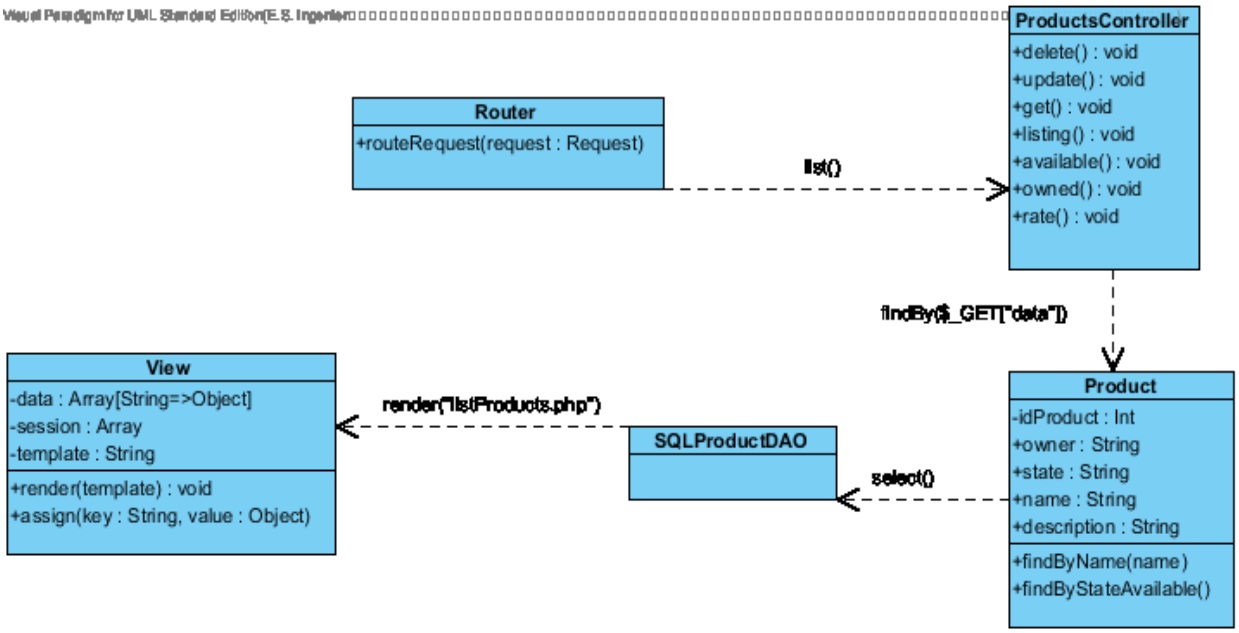
Añadir artículo



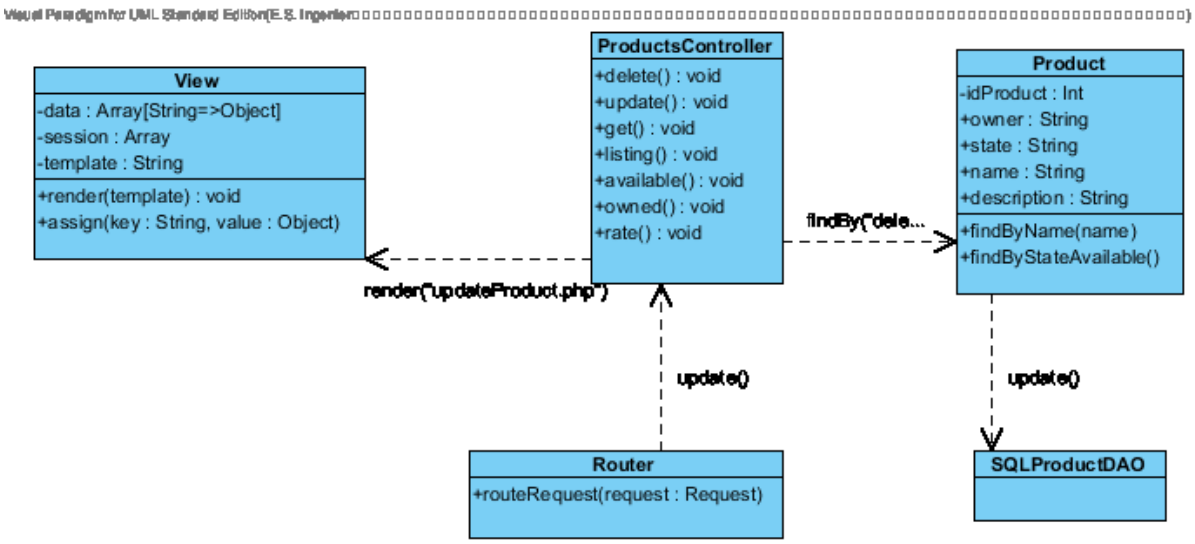
Listar artículos



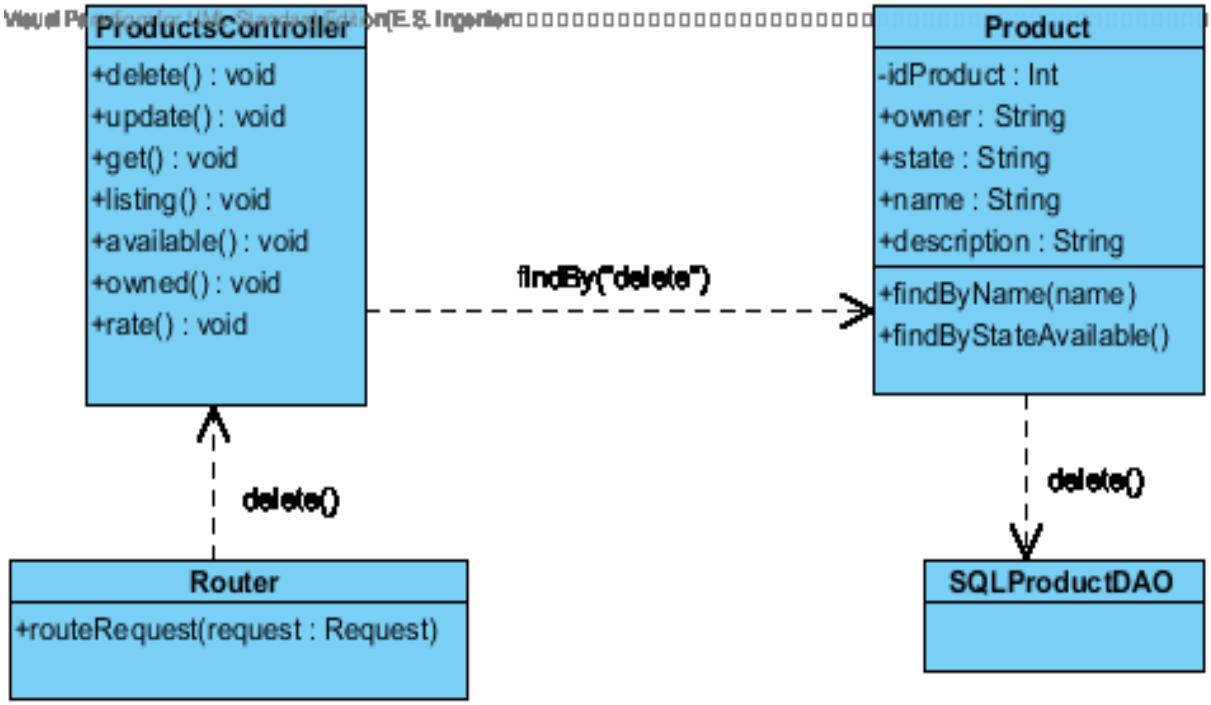
Buscar artículo



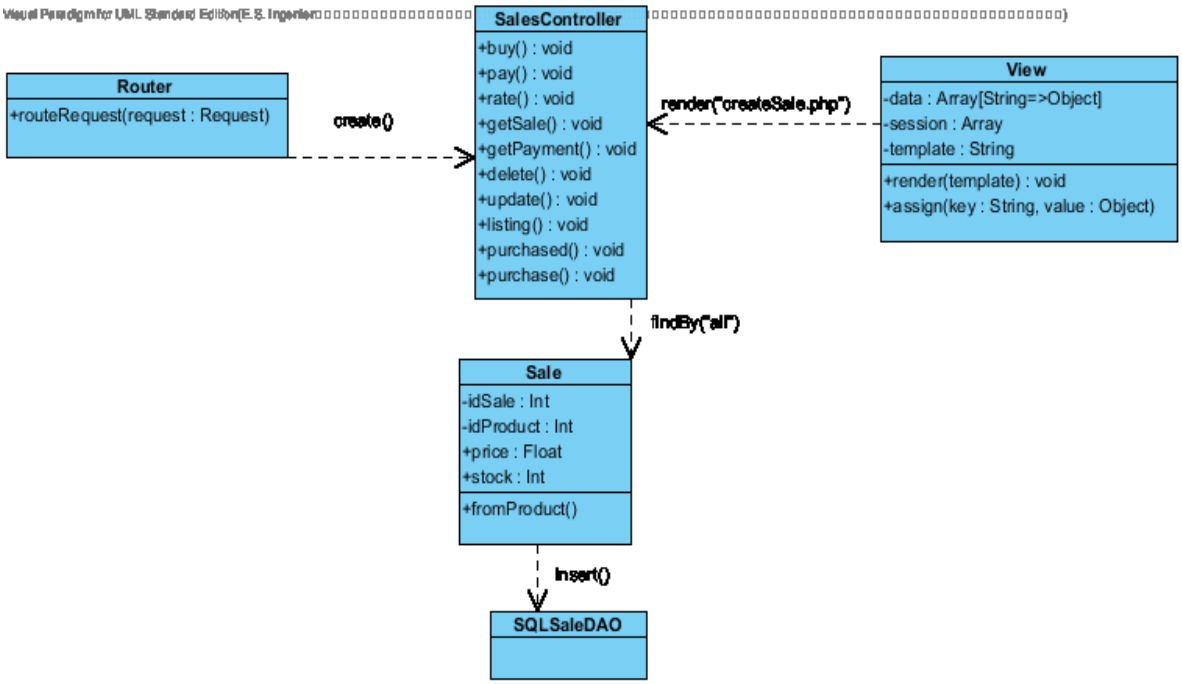
Modificar artículo



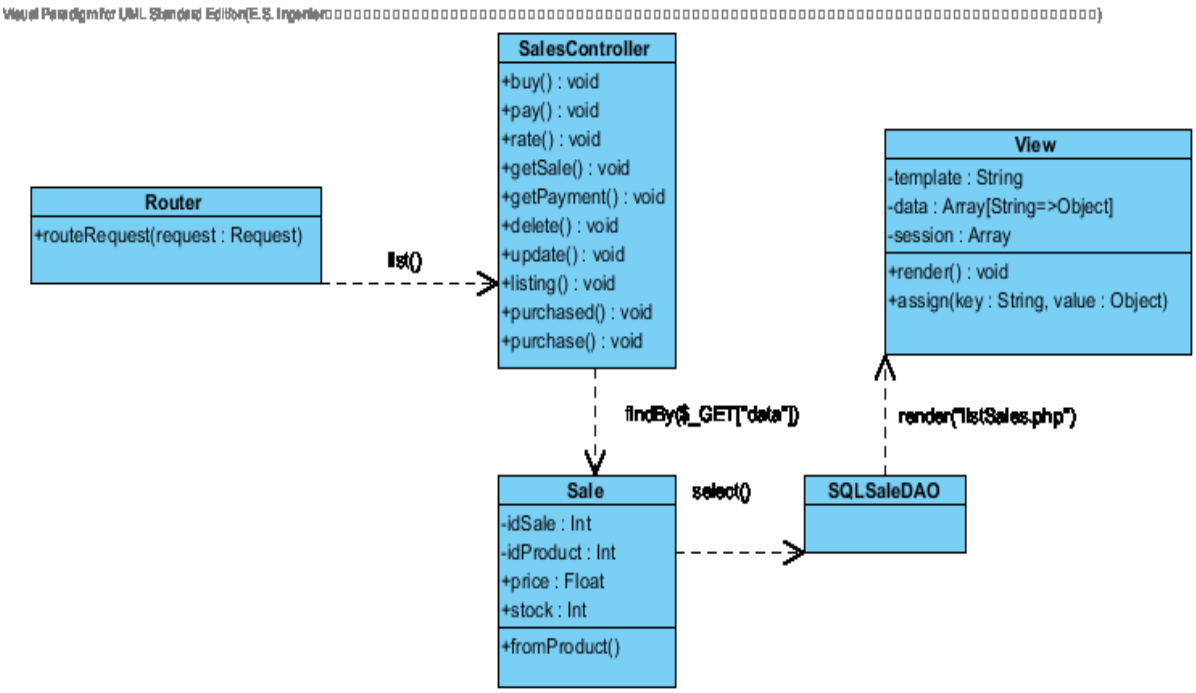
Eliminar artículo



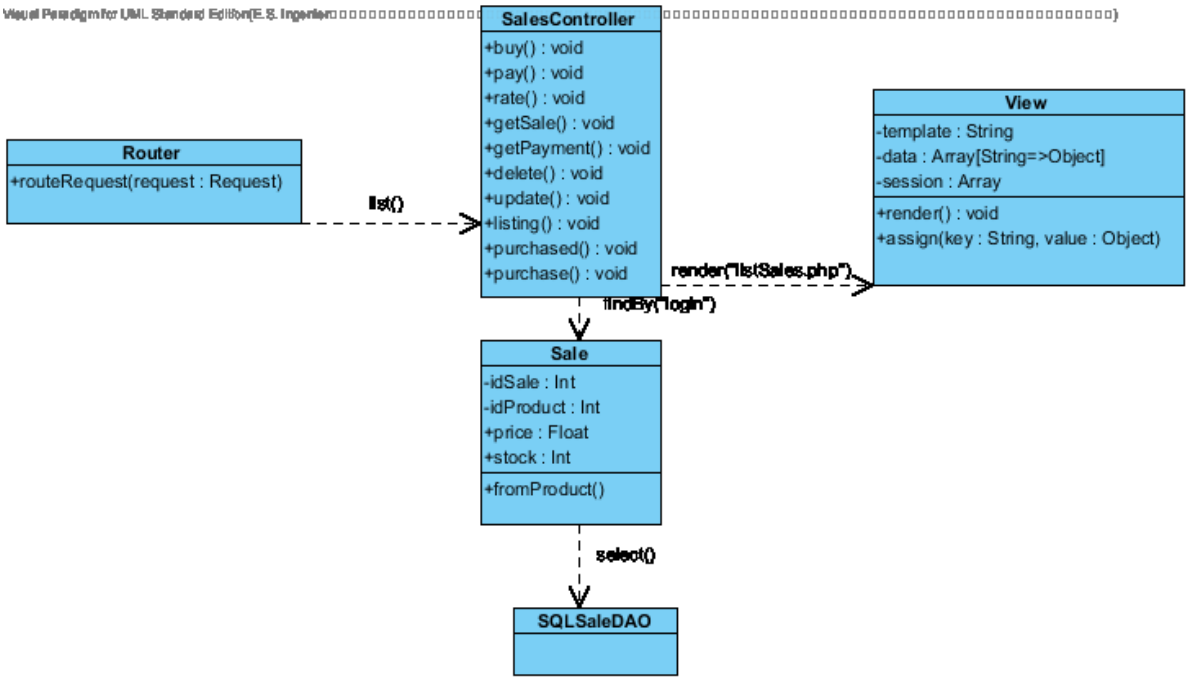
Añadir venta



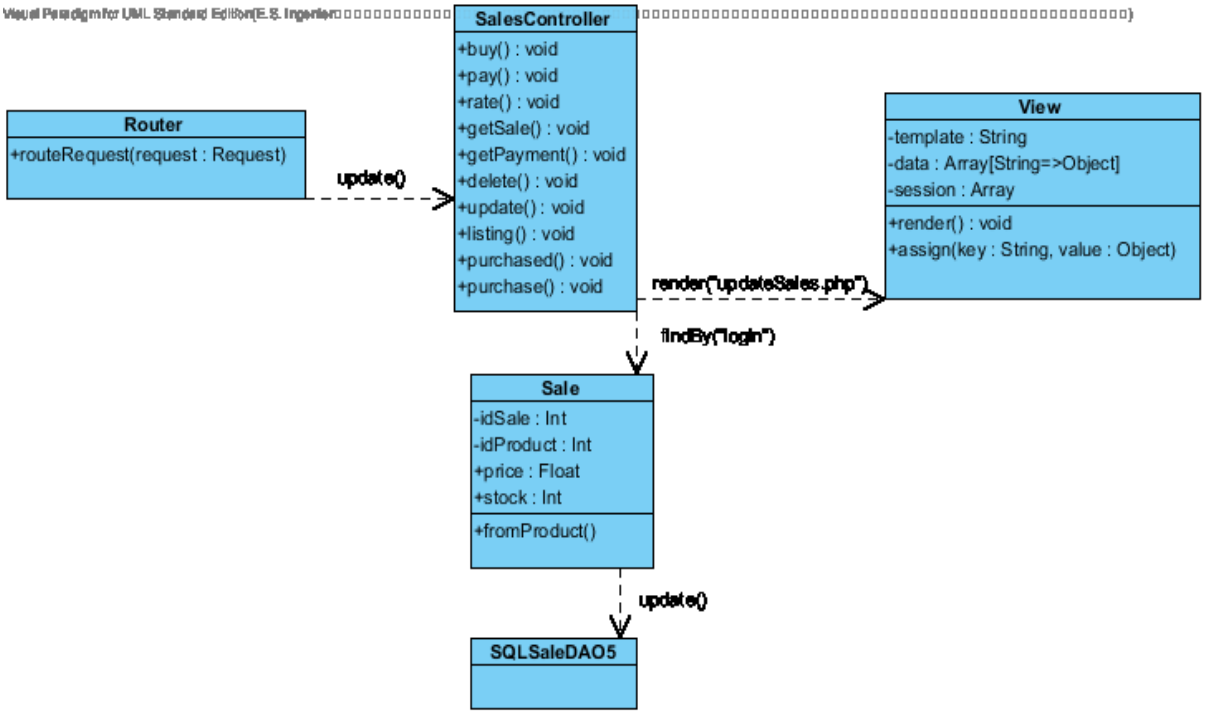
Buscar venta



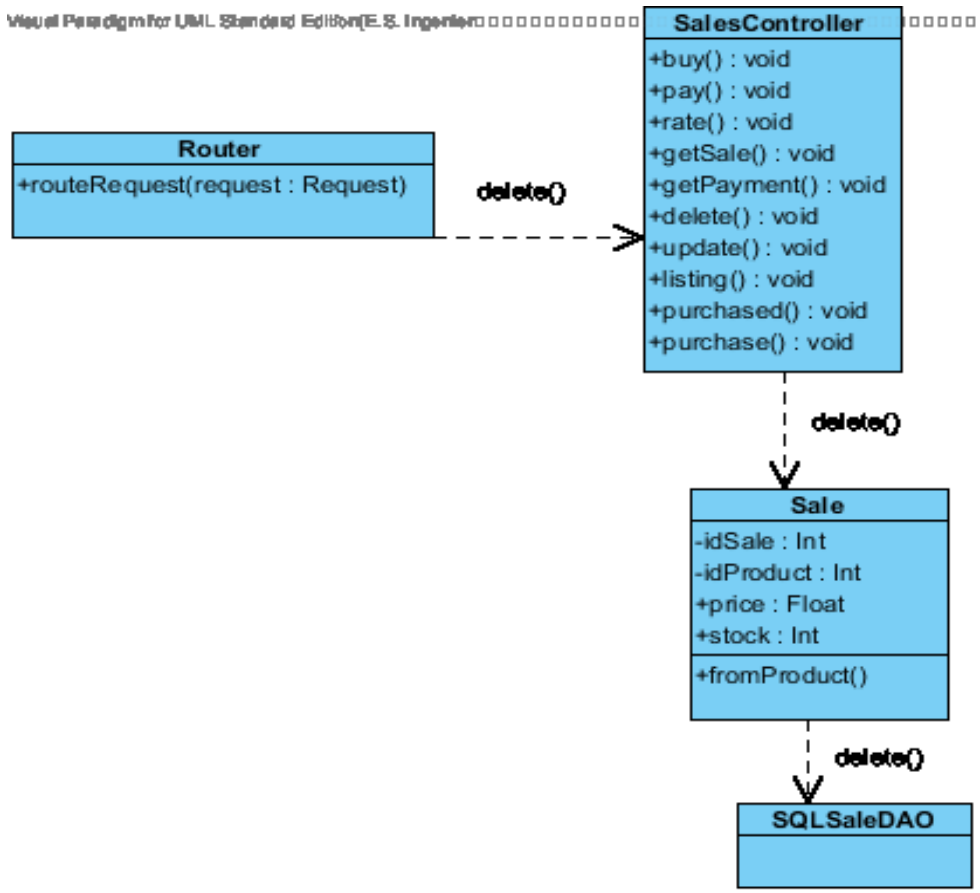
Listar ventas



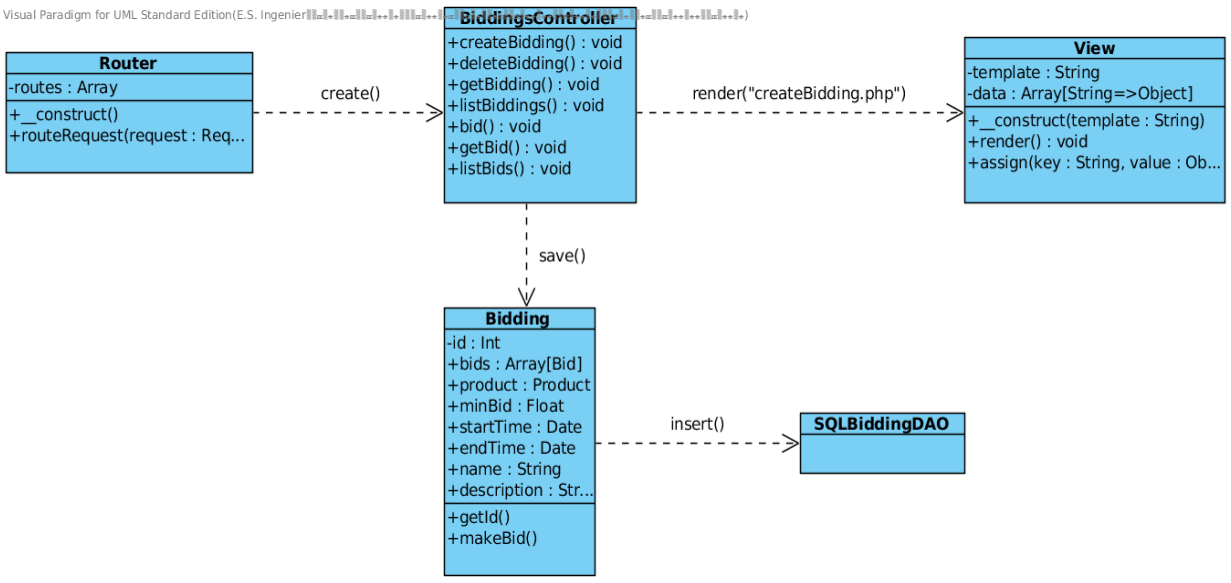
Modificar venta



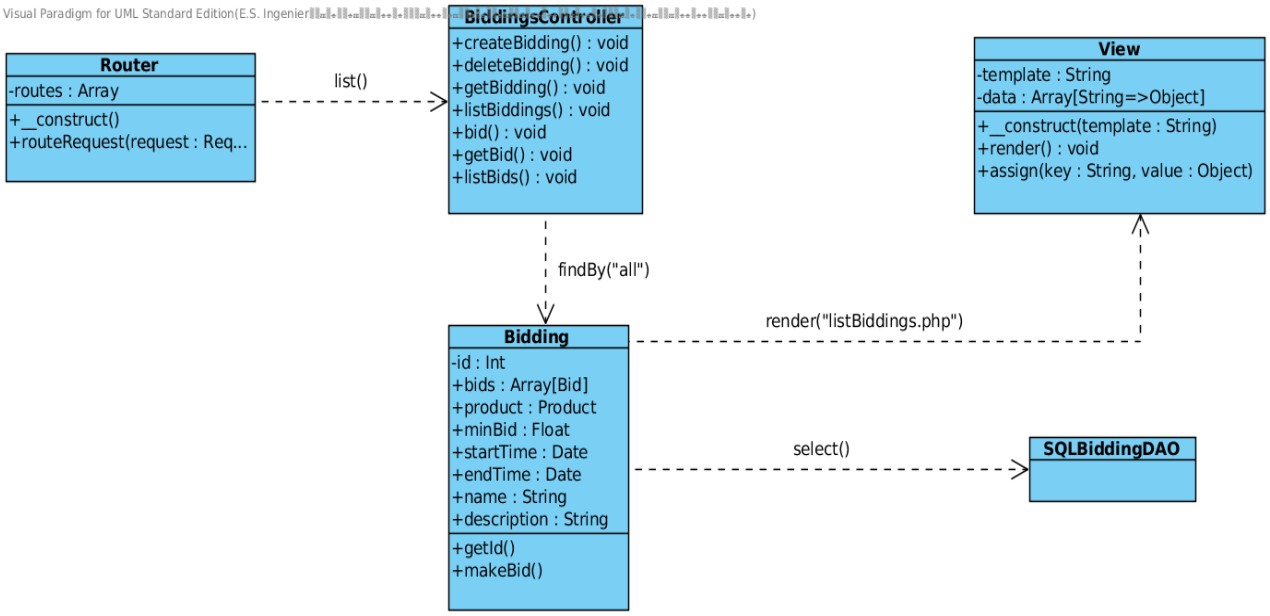
Eliminar venta



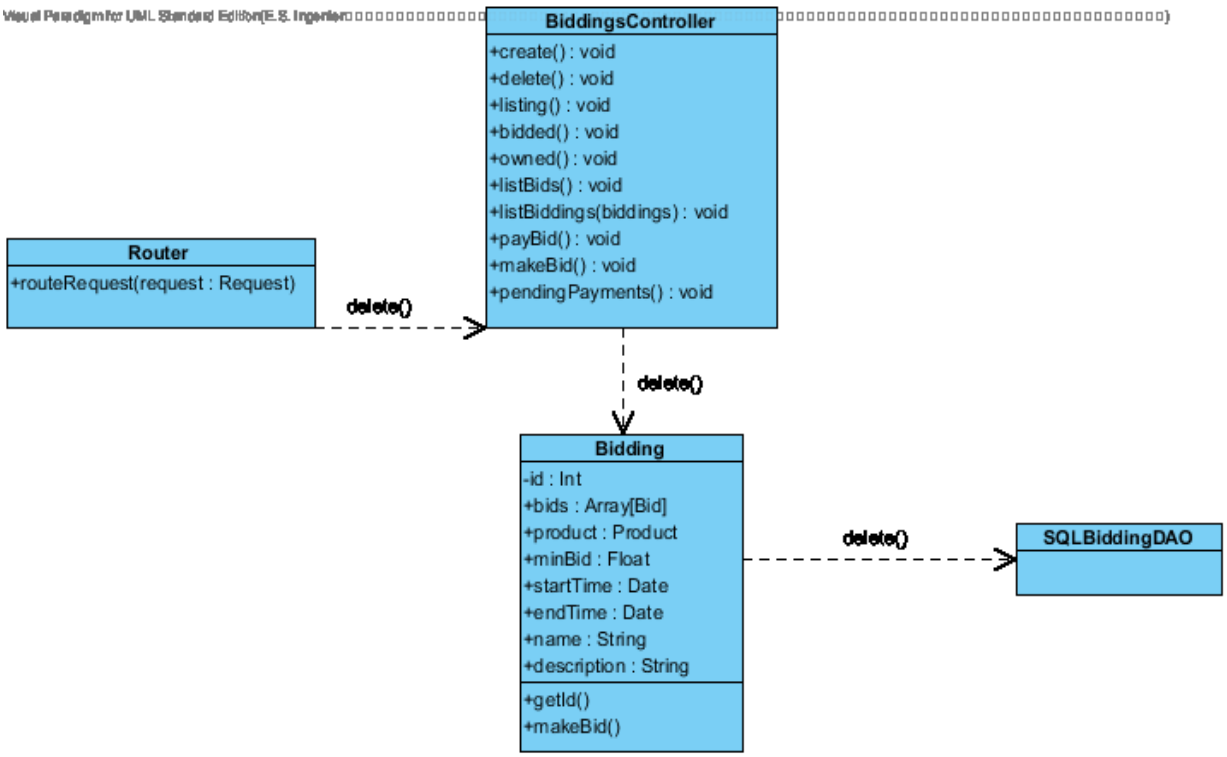
Añadir subasta



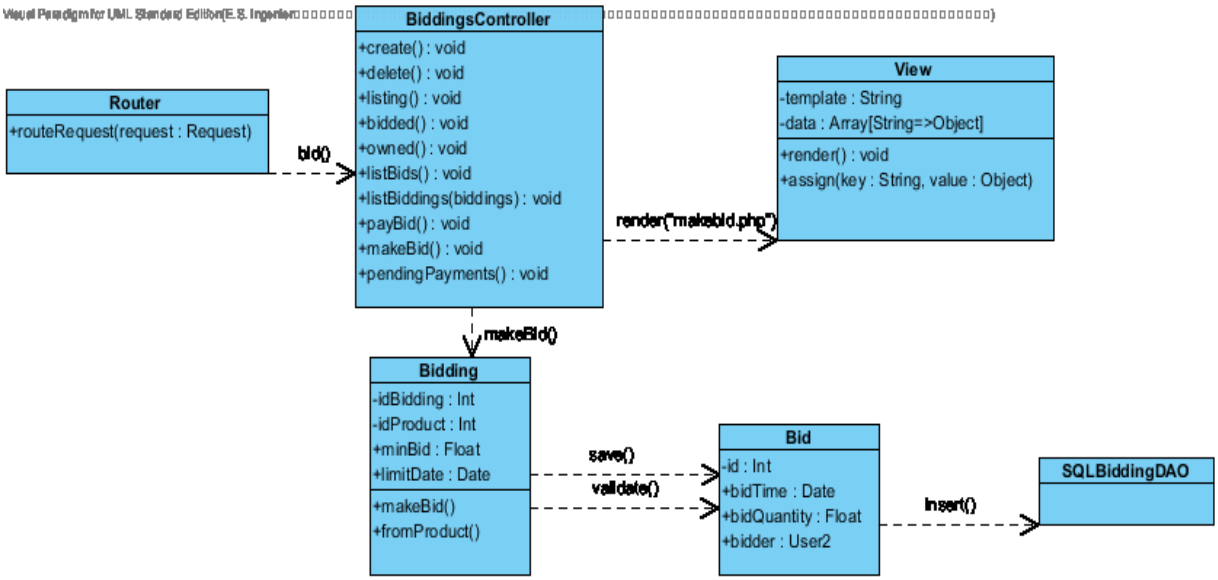
Listar subastas



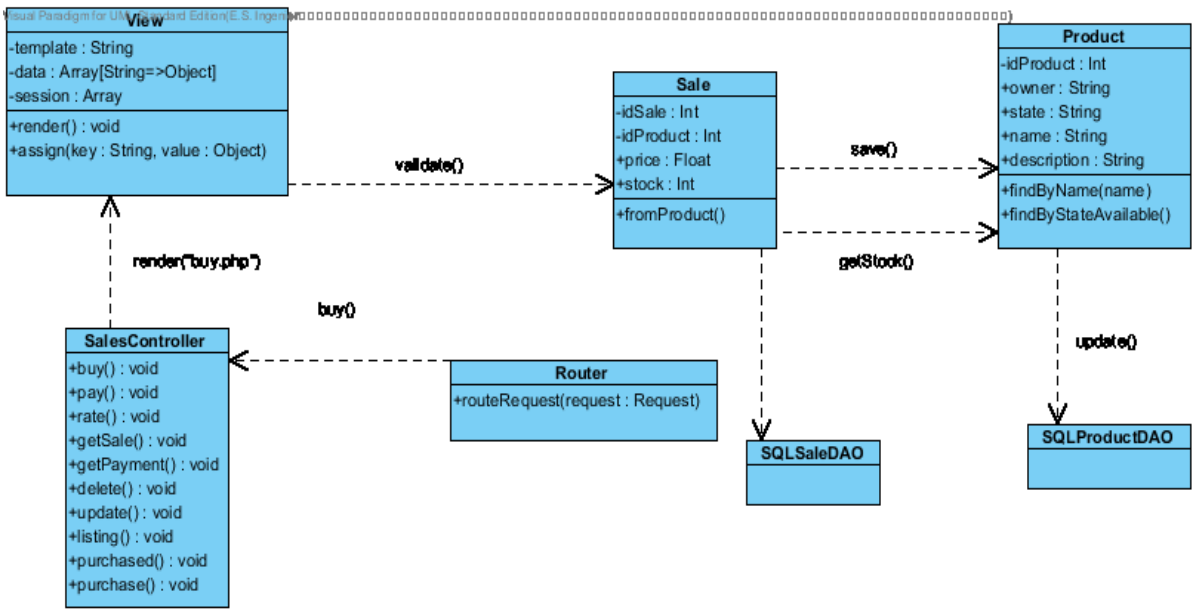
Eliminar subastas



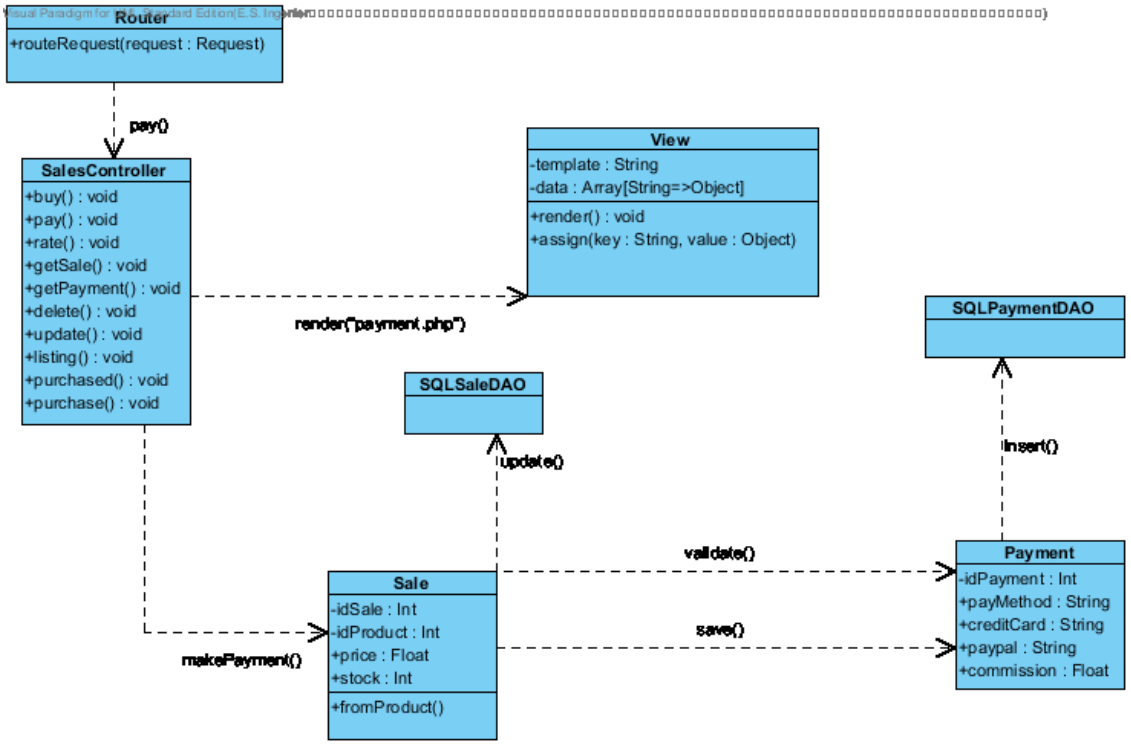
Pujar



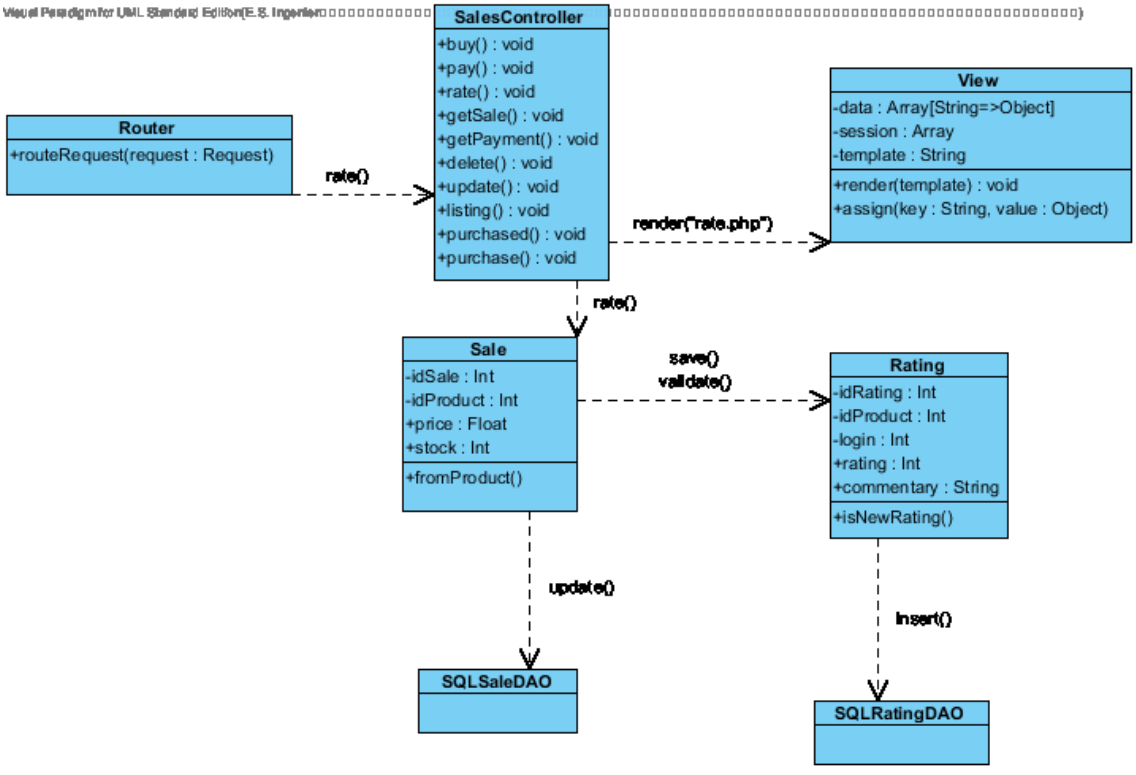
Comprar



Realizar pago



Calificar venta



Cambiar porcentajes

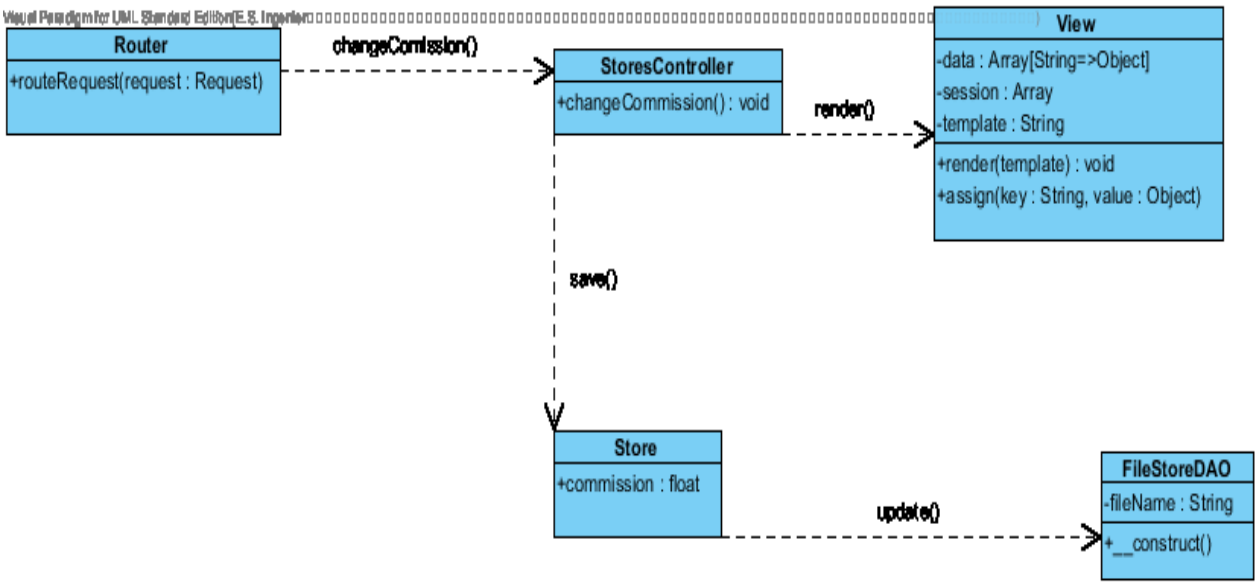
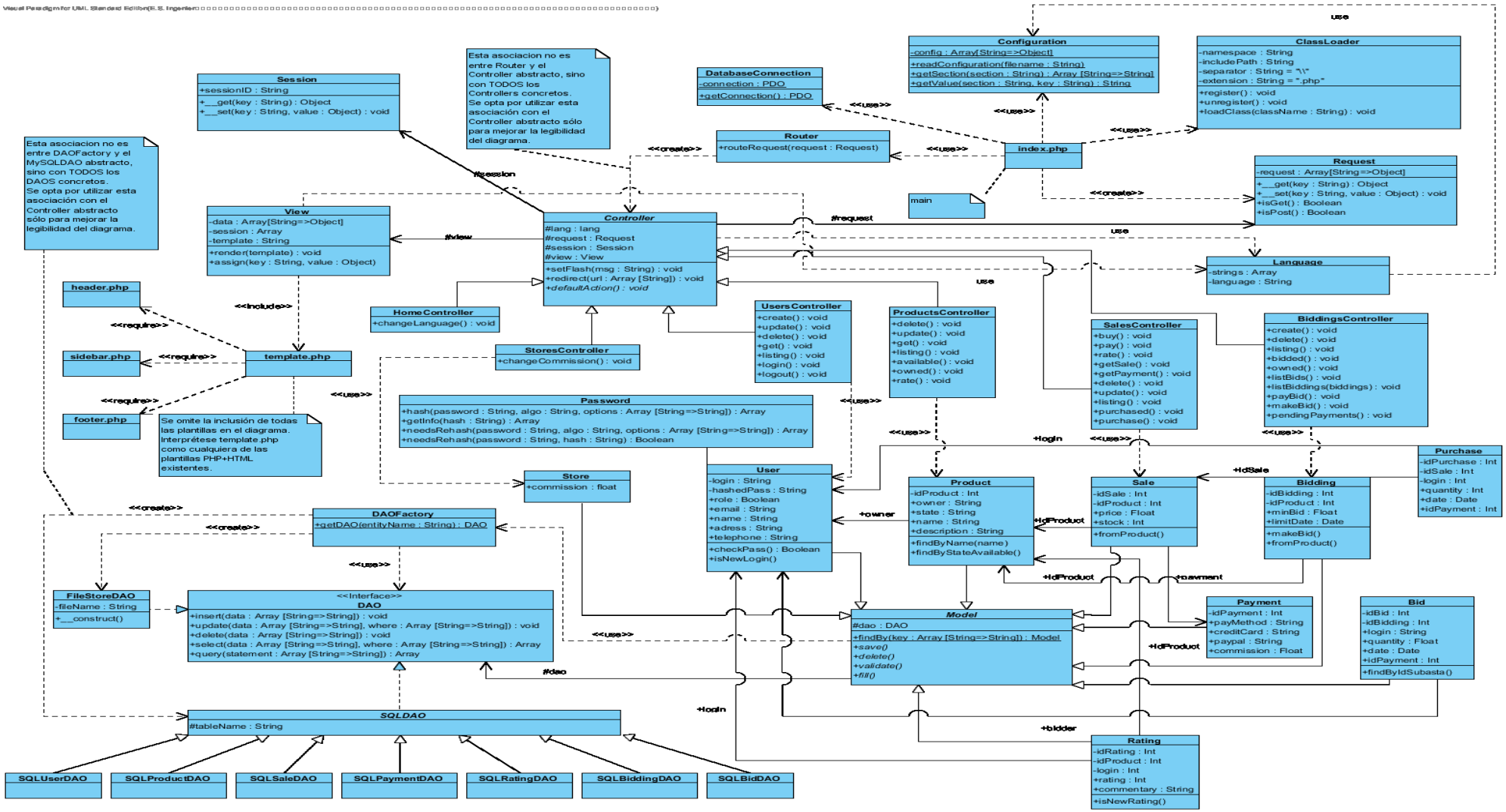


Diagrama de clases total



Documentación BD

Diagrama Entidad-Relación

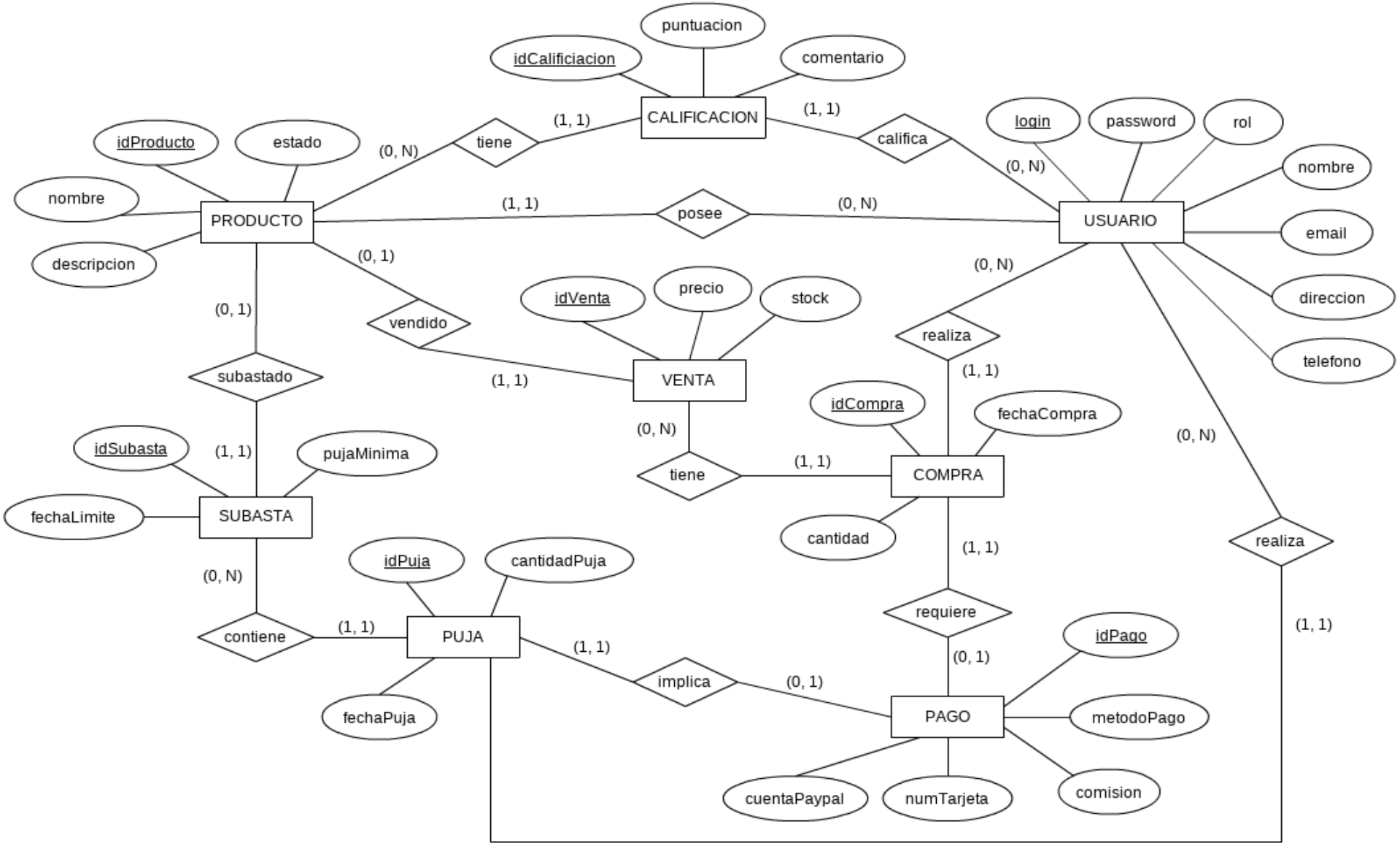
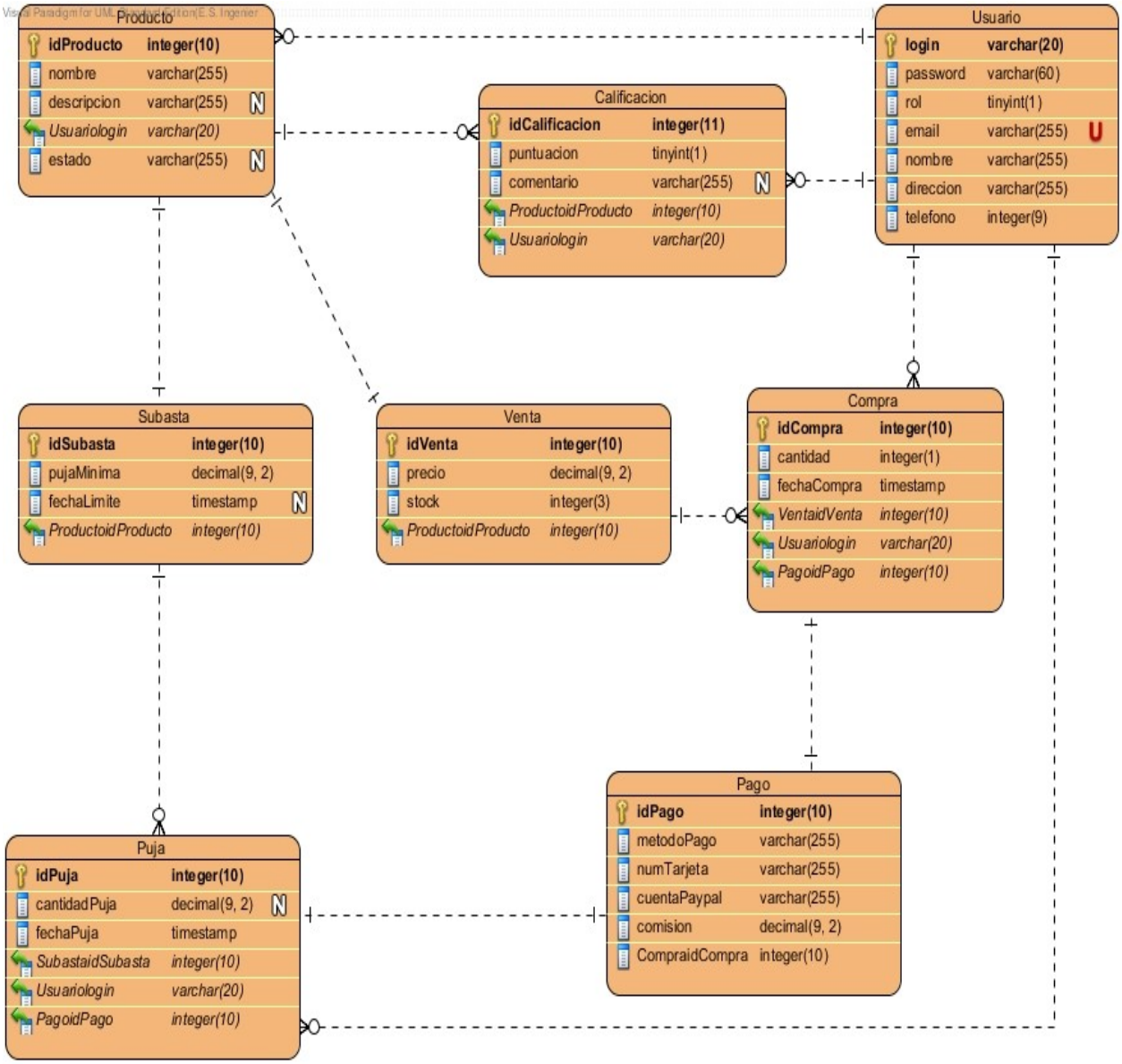


Diagrama de datos de la BD



Estructura física

Atributo	Estructura	Descripción
cantidad	int(10)	Identificador único de pago. Auto incrementado.
cantidadPuja	decimal(9,2)	Cantidad (en euros) de la puja realizada
comentario	Text	Comentario dado junto a la puntuación. Puede ser nulo.
comisión	decimal(9,2)	Comisión recibida por la empresa tras la realización del pago
cuentaPaypal	varchar(255)	Cuenta de PayPal del usuario
descripción	Text	Descripción del producto
dirección	varchar(255)	Dirección postal (completa) del usuario.
email	varchar(255)	Email del usuario, único (ie, no puede haber dos usuarios con el mismo email)
estado	enum('pendiente', 'subasta','venta')	Estado del producto: en venta, en subasta o aun no establecido (pendiente).
fechaCompra	Timestamp	Fecha y hora (TIMESTAMP) del momento de realización de la compra. Auto generado por la BD en el momento de inserción.
fechaLimite	Timestamp	Fecha y hora (TIMESTAMP) límite para la subasta. Puede ser nulo
fechaPuja	Timestamp	Fecha y hora (TIMESTAMP) de la realización de la puja. Se escribe automáticamente en el instante de inserción en la BD.
idCalificacion	int(11)	Identificador único de calificación. Auto incrementado.
idCompra	int(10)	Identificador único de compra. Auto incrementado.
idPago	int(10)	Identificador del pago asociado a esta compra. Puede ser nulo mientras la compra no haya sido pagada.
idProducto	int(10)	Identificador único de producto. Auto incrementado.
idPuja	int(10)	Identificador único de pujas. Auto incrementado.
idSubasta	int(10)	Identificador de la subasta donde se realiza la puja (clave foránea a SUBASTA.idSubasta). Puede ser nulo para permitir almacenamiento de pujas de subastas eliminadas (un "SET NULL").
idVenta	int(10)	Identificador único de ventas. Auto incrementado.
Login	varchar(20)	Login del usuario

metodoPago	enum(paypal, tarjeta)	Método de pago establecido. Puede ser "tarjeta" para referirse a tarjeta de crédito o "PayPal" para realización de pagos a través de PayPal.
nombre	varchar(255)	Nombre del producto
numTarjeta	varchar(255)	Número de tarjeta de crédito. Puede ser nulo, debería tener valor solamente si dicho método ha sido seleccionado.
password	varchar(60)	Password del usuario, almacenada con un hash Bcrypt
precio	decimal(9,2)	Precio de venta del producto (en euros).
pujaMinima	decimal(9,2)	Puja mínima para la subasta (en euros). Dos valores decimales admitidos.
puntuacion	tinyint(1)	Puntuación dada por el usuario al producto. La base de datos lo limita de 0 a 255, la lógica de la aplicación deberá velar por almacenar valores correctos.
rol	num('admin', 'usuario')	Rol del usuario (Administrador o Usuario normal)
stock	int(3)	Cantidad de ejemplares del producto disponibles para venta.
telefono	int(9)	Teléfono de contacto (fijo o móvil) del usuario. Puede ser nulo.

Diccionario de datos

Atributo	TipoDeClave	Tabla	Descripción
cantidad	----------	Compra	Identificador único de pago. Auto incrementado.
cantidadPuja	----------	Puja	Cantidad (en euros) de la puja realizada
comentario	----------	Calificacion	Comentario dado junto a la puntuación. Puede ser nulo.
comisión	----------	Pago	Comisión recibida por la empresa tras la realización del pago
cuentaPaypal	----------	Pago	Cuenta de PayPal del usuario
descripción	----------	Producto	Descripción del producto
dirección	----------	Usuario	Dirección postal (completa) del usuario.
email	----------	Usuario	Email del usuario, único (ie, no puede haber dos usuarios con el mismo email)
estado	----------	Producto	Estado del producto: en venta, en subasta o aun no establecido (pendiente).
fechaCompra	----------	Compra	Fecha y hora (TIMESTAMP) del momento de realización de la compra. Auto generado por la BD en el momento de inserción.
fechaLimite	----------	Subasta	Fecha y hora (TIMESTAMP) límite para la subasta. Puede ser nulo
fechaPuja	----------	Puja	Fecha y hora (TIMESTAMP) de la realización de la puja. Se escribe automáticamente en el instante de inserción en la BD.
idCalificacion	PK(Calificacion)	Calificaciona	Identificador único de calificación. Auto incrementado.
idCompra	PK(Compra)	Compra	Identificador único de compra. Auto incrementado.
idPago	PK(Pago) FK(Compra) FK(Puja)	Pago	Identificador del pago asociado a esta compra. Puede ser nulo mientras la compra no haya sido pagada.
idProducto	PK(Producto) FK(Calificacion) FK(Subasta) FK(Venta)	Producto	Identificador único de producto. Auto incrementado.
idPuja	PK(Puja)	Puja	Identificador único de pujas. Auto incrementado.
idSubasta	PK(Subasta)	Subasta	Identificador de la subasta donde se realiza la puja (clave foránea a

	FK(Producto) FK(Puja)		SUBASTA.idSubasta). Puede ser nulo para permitir almacenamiento de pujas de subastas eliminadas (un "SET NULL").
idVenta	PK(Venta) FK(Compra)	Venta	Identificador único de ventas. Auto incrementado.
login	PK(Usuario) FK(Calificacion) FK(Compra) FK(Puja)	Usuario	Login del usuario
metodoPago	----------	Pago	Método de pago establecido. Puede ser "tarjeta" para referirse a tarjeta de crédito o "PayPal" para realización de pagos a través de PayPal.
nombre	----------	Producto	Nombre del producto
numTarjeta	----------	Pago	Número de tarjeta de crédito. Puede ser nulo, debería tener valor solamente si dicho método ha sido seleccionado.
password	----------	Usuario	Password del usuario, almacenada con un hash Bcrypt
precio	----------	Venta	Precio de venta del producto (en euros).
pujaMinima	----------	Subasta	Puja mínima para la subasta (en euros). Dos valores decimales admitidos.
puntuacion	----------	Calificacion	Puntuación dada por el usuario al producto. La base de datos lo limita de 0 a 255, la lógica de la aplicación deberá velar por almacenar valores correctos.
rol	----------	Usuario	Rol del usuario (Administrador o Usuario normal)
stock	----------	Venta	Cantidad de ejemplares del producto disponibles para venta.
telefono	----------	Usuario	Teléfono de contacto (fijo o móvil) del usuario. Puede ser nulo.