



PROYECTO FINAL
ASIGNATURA: PROGRAMACIÓN
PROFESOR: PACO ALDARIAS RAYA
CENTRO EDUCATIVO: CEED <http://www.ceedcv.es>
CURSO: 2013/14

[prg1314finalgrupo1](#)
ALUMNO/S:
Francisco Aldarias Raya
paco.aldarias@ceedcv.es

TITULO:
Tienda Online

FECHA:
06/05/14

CVS:
<https://bitbucket.org/ceeddaw/prg1314finalgrupo1>

NOTA: ESTO ES UNA PLANTILLA. DEBEREIS COGER ESTE ARCHIVO Y RELLENAR LOS DATOS DE VUESTRA APLICACIÓN.

ÍNDICE:

Índice de contenido

1INTRODUCCIÓN.....	3
2DISEÑO DE LA APLICACIÓN.....	3
2.1DIAGRAMA DE CLASES.....	3
2.2DISEÑO FÍSICO DE CLASES EN JAVA.....	4
2.3DIAGRAMA DE PAQUETES.....	5
2.4DESCRIPCIÓN DEL MVC.....	5
3DISEÑO DE LA BASE DE DATOS.....	5
3.1ANÁLISIS DE REQUISITOS.....	5
3.2DISEÑO CONCEPTUAL ENTIDAD-RELACIÓN.....	6
3.3DISEÑO LÓGICO RELACIONAL.....	7
3.4DISEÑO FÍSICO.....	7
3.4.1DISEÑO DE LA BASE DE DATOS.....	7
3.4.2TABLAS EN MYSQL.....	8
4DISEÑO FISICO DE CLASES EN JAVA.....	10
5CAPTURAS DE PANTALLA.....	11
5.1Login.....	11
6SOFTWARE UTILIZADO PARA EL DESARROLLO.....	12
7ESTRUCTURA PAQUETES EN NETBEANS.....	12
8DESPLIEGUE.....	13
8.1Mysql.....	13
9REPARTO DEL PROYECTO.....	13
10CONTROL DE VERSIONES.....	14
11PROBLEMAS ENCONTRADOS.....	14
12MEJORAS.....	14
13BIBLIOGRAFIA.....	14

1 INTRODUCCIÓN

La aplicación realiza la tienda online. Se realizará la compra de productos por parte de cliente.

La aplicación esta diseñada utilizando el MVC. Permite guardar la información en ficheros, base de datos relacionales y en base de datos orientada a objetos.

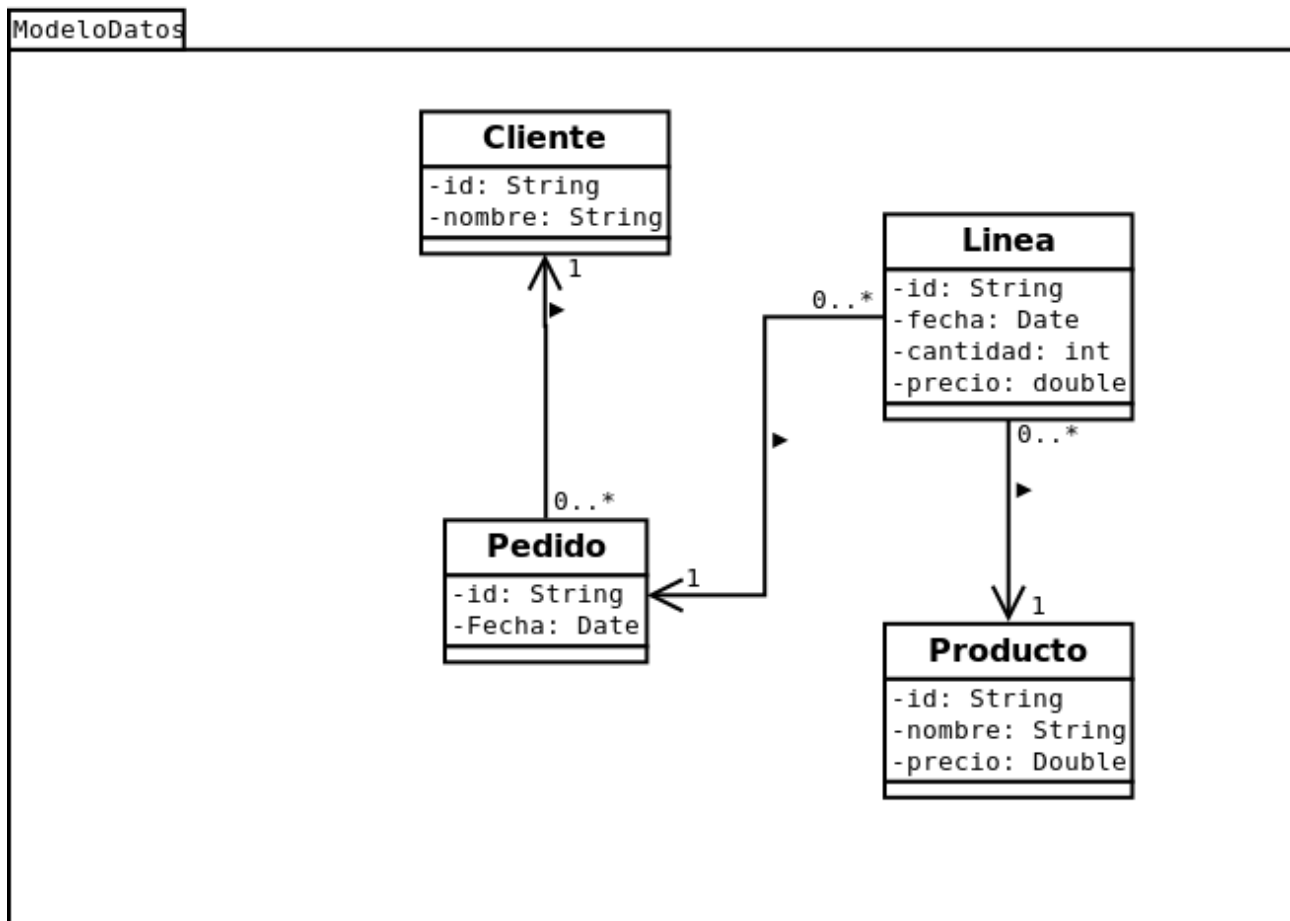
Se dispondrán de una serie de artículos y de clientes registrados que realizaran compras, las cuales se almacenaran en un carrito.

2 DISEÑO DE LA APLICACIÓN

2.1 DIAGRAMA DE CLASES

El diagrama clases es:

DIAGRAMA DE CLASES
TIENDA



Nota:

Se han omitido las funciones set y get de las clases para que sea más legible el gráfico.

2.2 DISEÑO FÍSICO DE CLASES EN JAVA.

Consultando el diagrama de clases podemos obtener las siguientes clases y sus atributos:

```
public class Cliente {  
    private String id;  
    private String nombre;  
    private String email;  
}
```

```
public class Pedido {  
    private String id;  
    private Cliente cliente;  
    private String fecha;  
}
```

```
public class Linea {  
    private String idlinea;  
    private Pedido pedido;  
    private String importe;  
    private String cantidad;  
    private String fecha;  
    private Producto producto;  
}
```

```
public class Producto {  
    private String id;  
    private String nombre;  
    private String precio;  
}
```

Nota:

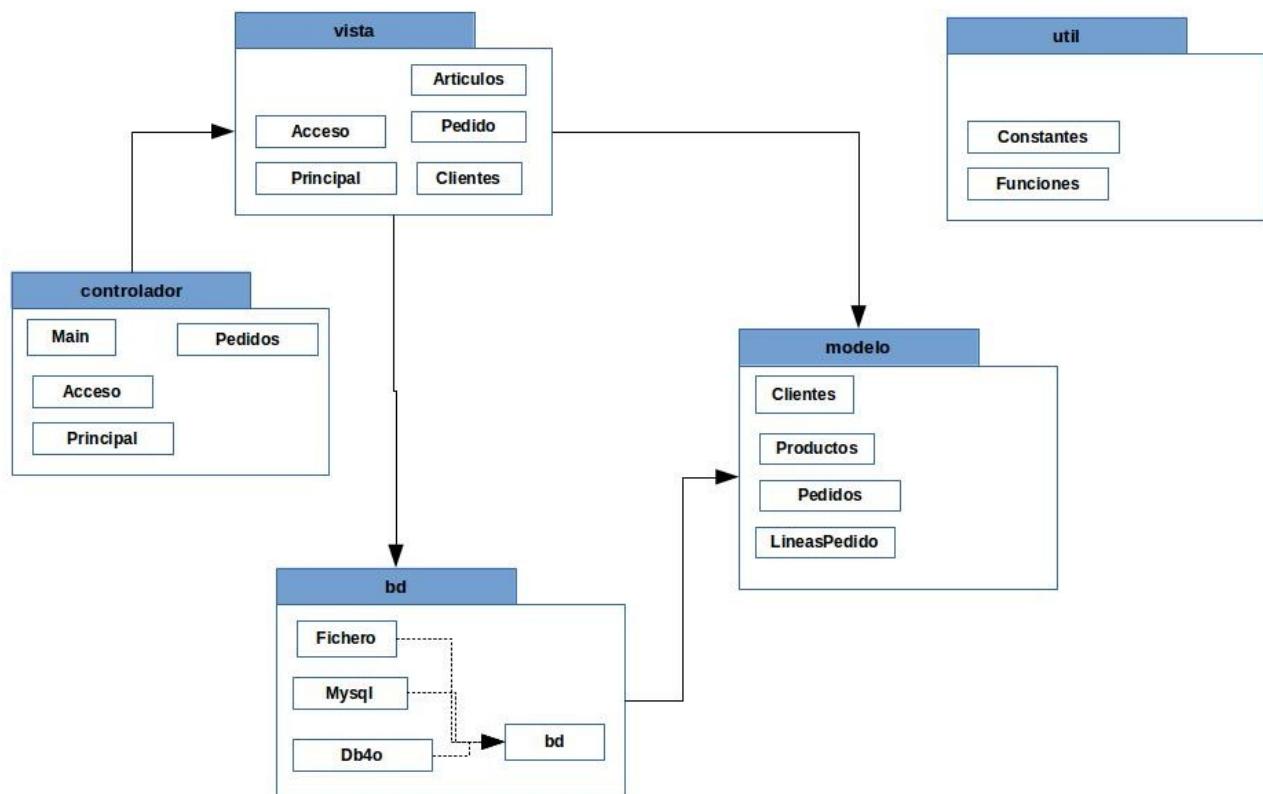
- Se puede apreciar que en la relación 1 a Muchos , en la parte 1 va un objeto.
- Para realizar las relaciones entre clases, no hay atributos que sean claves

ajenas, sino objetos de una clase.

2.3 DIAGRAMA DE PAQUETES

El diagrama de paquetes es:

DIAGRAMA PAQUETES. TIENDA
Paco Aldarias. 14/02/14



2.4 DESCRIPCIÓN DEL MVC

La siguiente aplicación cumple el diseño MVC (Modelo - Vista - Controlador). Para ello se crean los paquetes siguientes:

- **Modelo:** Contiene todas las clases de objetos que se desean guardar información. En este caso coincide con las tablas del diseño Entidad Relación.
- **Vista:** Contienen las clases que interactúan con el usuario. Se ha extraído todo el código posible de las vistas para ponerlo en controlador.
- **Controlador.** Están las clases que

3 DISEÑO DE LA BASE DE DATOS

3.1 ANÁLISIS DE REQUISITOS.

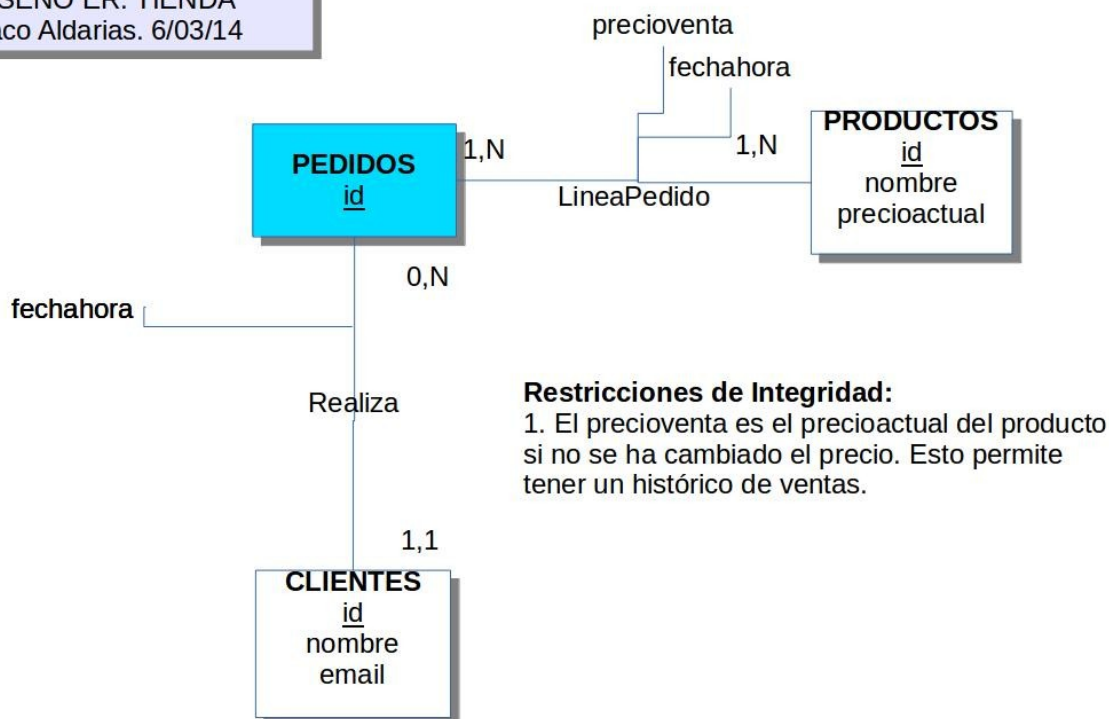
Se desea realizar la gestión de ventas de una tienda. Para ellos se grabarán los pedidos realizados de un cliente de determinados productos.

Restricciones de cardinalidad de las relaciones:

- Un cliente puede tener 0 o muchos pedidos.
- Un pedido corresponde a un cliente y sólo uno. Y debe tener 1 o muchos líneas de pedido.
- Una línea de pedido debe siempre un pedido y sólo uno. Y un producto y sólo 1.
- Un producto puede estar en muchas líneas de pedido o ninguna.

3.2 DISEÑO CONCEPTUAL ENTIDAD-RELACIÓN

DISEÑO ER. TIENDA
Paco Aldarias. 6/03/14



Restricciones de Integridad:

1. El precio venta es el precio actual del producto si no se ha cambiado el precio. Esto permite tener un histórico de ventas.

Aclaraciones:

Todas las tablas tienen como clave primaria el atributo id, que es número autoincremento. Eso simplifica la elección de claves primarias.

En líneas de pedido los atributos precio y fecha son los de la tabla pedidos, con objeto de que si en un futuro se cambian los precios de los artículos quede constancia de ello.

Restricciones de Integridad:

1. El precio actual del producto coincide con el precio venta de línea de pedido en el momento de la venta. Pero el precio venta puede cambiar.
2. La fecha de pedido es la fecha que se crea el pedido, y la fecha de línea es la fecha que se introduce la línea de pedido que inicialmente será la misma.

3.3 DISEÑO LÓGICO RELACIONAL.

Las relaciones son:

1. CLIENTES (idcliente, nombrecliente, email)
CP:id
2. PRODUCTOS(idproducto, nombreproducto,precioactual)
CP:idproducto
3. PEDIDO (idpedido, idcliente, fechapedido)
CP: idpedido
VNN: idcliente
CA: idcliente → CLIENTE
4. LINEAPEDIDO (idlinea, idpedido, idarticulo, cantidad, precioventa, fechalinea)
CP:idlinea
VNN: idpedido
VNN: idarticulo
CA:idpedido-->PEDIDO
CA:idarticulo-->ARTICULO

NOTA:

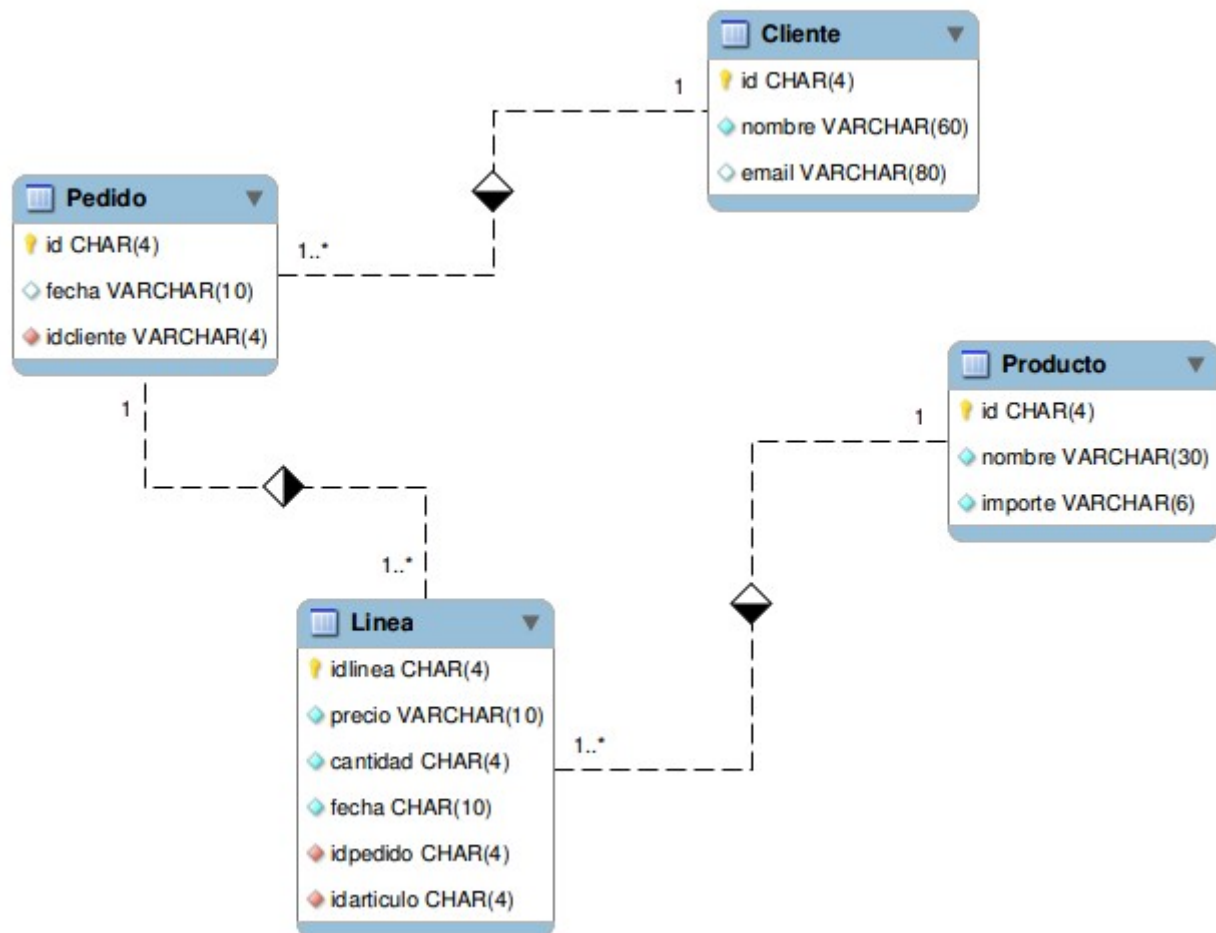
- CP: Indica Clave Primaria
- CA: Indica Clave Ajena

Todas las relaciones están en 3ra Forma Normal.

3.4 DISEÑO FÍSICO.

El diseño físico se va a realizar con el SGBD Mysql.

3.4.1 DISEÑO DE LA BASE DE DATOS.



3.4.2 TABLAS EN MYSQL.

El diseño físico se va a realizar sobre la base de datos relacional MYSQL. Se encuentra en el fichero prg1314finalgrupo1.sql. Se deberá importar una vez creada la base de datos.

```
--
-- Base de datos: `prg1314finalgrupo1`
--

--
-- Estructura de tabla para la tabla `Cliente`
--

CREATE TABLE IF NOT EXISTS `Cliente` (
  `id` char(4) NOT NULL,
```



```
`nombre` varchar(60) NOT NULL DEFAULT "",
`email` varchar(80) DEFAULT NULL,
PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Volcado de datos para la tabla `Cliente`
--

INSERT INTO `Cliente` (`id`, `nombre`, `email`) VALUES
('1', 'a', 'a');

-- -----

--
-- Estructura de tabla para la tabla `Linea`
--

CREATE TABLE IF NOT EXISTS `Linea` (
  `idlinea` char(4) NOT NULL,
  `importe` varchar(10) DEFAULT NULL,
  `cantidad` char(4) DEFAULT NULL,
  `fecha` char(10) DEFAULT NULL,
  `idpedido` char(4) DEFAULT NULL,
  `idproducto` char(4) DEFAULT NULL,
  PRIMARY KEY (`idlinea`),
  KEY `idpedido` (`idpedido`),
  KEY `idproducto` (`idproducto`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- -----

--
-- Estructura de tabla para la tabla `Pedido`
--

CREATE TABLE IF NOT EXISTS `Pedido` (
  `id` char(4) NOT NULL,
  `fecha` varchar(10) DEFAULT NULL,
  `idcliente` varchar(4) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `idcliente` (`idcliente`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Volcado de datos para la tabla `Pedido`
--

INSERT INTO `Pedido` (`id`, `fecha`, `idcliente`) VALUES
('1', '2014-4-9', '1');
```

```
-- -----  
--  
-- Estructura de tabla para la tabla `Producto`  
--  
CREATE TABLE IF NOT EXISTS `Producto` (  
  `idproducto` char(4) NOT NULL,  
  `nombre` varchar(30) NOT NULL DEFAULT "",  
  `precio` varchar(6) NOT NULL DEFAULT '0',  
  PRIMARY KEY (`idproducto`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
--  
-- Restricciones para tablas volcadas  
--  
  
--  
-- Filtros para la tabla `Linea`  
--  
ALTER TABLE `Linea`  
  ADD CONSTRAINT `Linea_ibfk_3` FOREIGN KEY (`idproducto`) REFERENCES  
  `Producto` (`idproducto`),  
  ADD CONSTRAINT `Linea_ibfk_2` FOREIGN KEY (`idpedido`) REFERENCES `Pedido`  
  (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION;  
  
--  
-- Filtros para la tabla `Pedido`  
--  
ALTER TABLE `Pedido`  
  ADD CONSTRAINT `Pedido_ibfk_1` FOREIGN KEY (`idcliente`) REFERENCES  
  `Cliente` (`id`);
```

4 DISEÑO FÍSICO DE CLASES EN JAVA.

Consultando el diagrama de clases podemos obtener las siguientes clases y sus atributos:

```
public class Cliente {  
    private String id;  
    private String nombre;  
    private String email;  
    private Set pedidos = new HashSet(0);  
}
```

```
public class Pedido {  
    private String id;  
    private Cliente cliente;  
    private String fecha;  
    private Set lineas = new HashSet(0);  
}
```

```
public class Linea {  
    private String idlinea;  
    private Pedido pedido;  
    private String importe;  
    private String cantidad;  
    private String fecha;  
    private Producto producto;  
}
```

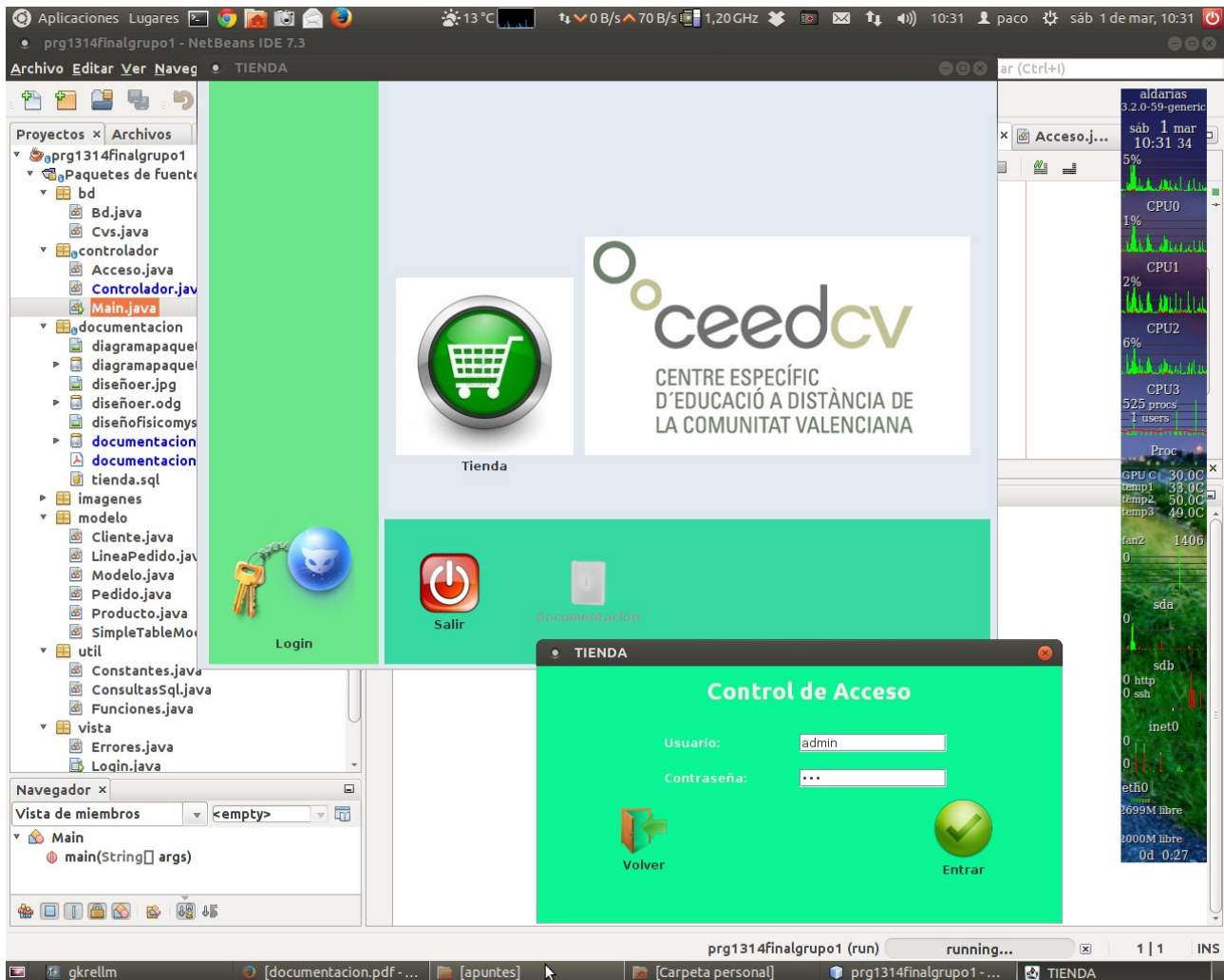
```
public class Producto {  
    private String id;  
    private String nombre;  
    private String precio;  
    private Set lineas = new HashSet(0);  
}
```

Se puede apreciar que en la relación 1 a Muchos , en la parte 1 va un objeto, y en la parte Muchos va una lista de objetos expresadas mediante set.

5 CAPTURAS DE PANTALLA

5.1 Login

Poniendo usuario: admin y contraseña: 123. Podemos acceder a la aplicación.

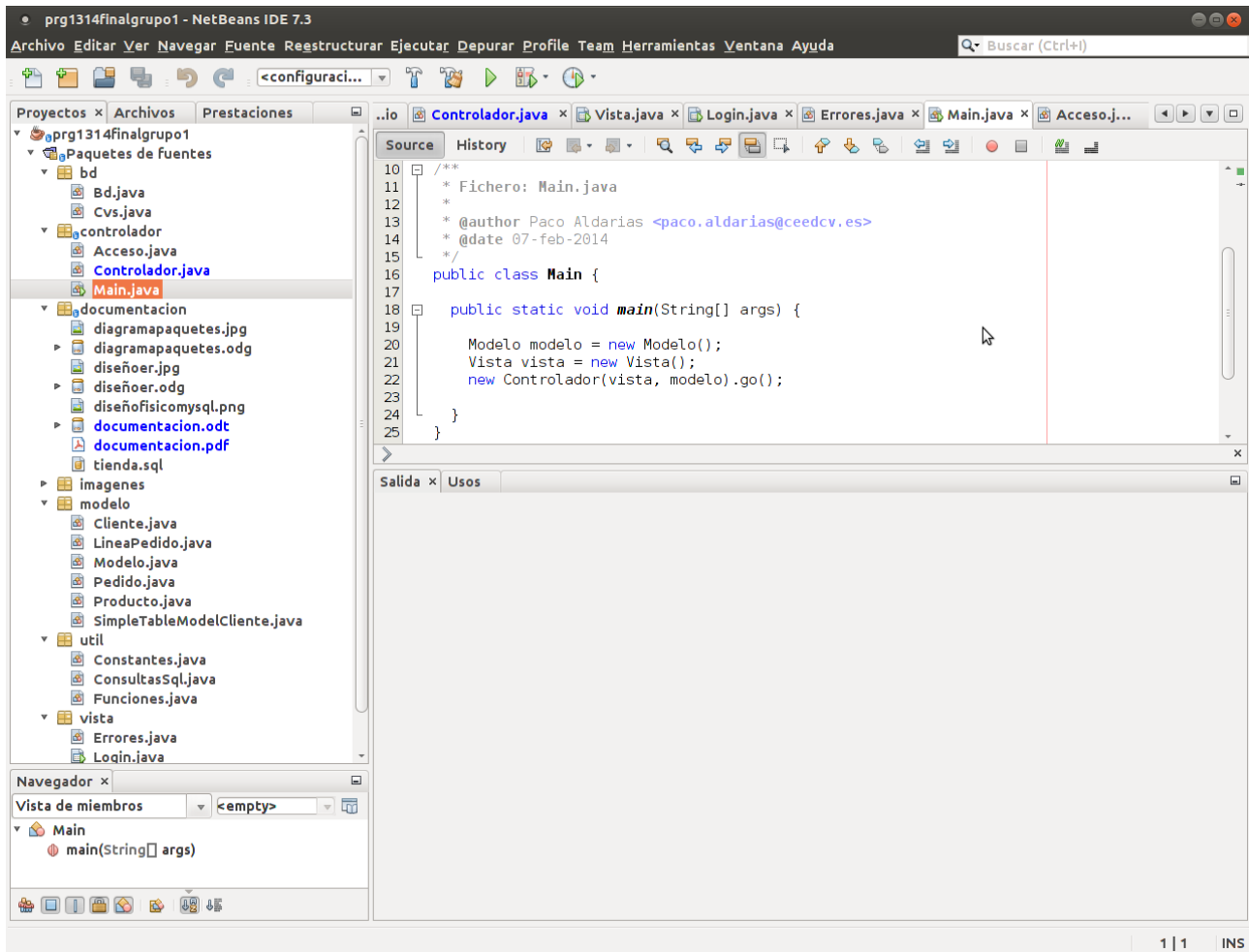


6 SOFTWARE UTILIZADO PARA EL DESARROLLO.

- Para el desarrollo de la aplicación ha necesitado de :
- Netbeans
 - Librerías JDK 1.7
 - Librerías de Db40
 - Mysql
 - Linux

7 ESTRUCTURA PAQUETES EN NETBEANS

Los paquetes que se han creado en el proyecto se pueden ver en la siguiente imagen.



8 DESPLIEGUE

Requiere tener instalado:

1. Mysql Versión del servidor: 5.5.37-0ubuntu0.12.04.1
2. Maquina Virtual Jva o JRE.

8.1 Mysql

Crear la base de datos **prg1314finalgrupo1** e importar el fichero [prg1314finalgrupo1.sql](#) que se encuentra en la carpeta documentación/sql. Poner como usuario: admin y contraseña: 123 de acceso a la base de datos.

9 REPARTO DEL PROYECTO.

Como sólo ha sido de una persona no ha existido. Habría que citar aquí la parte que ha hecho cada

uno.

Una forma de hacerlo sería en hacer en conjunto el diseño del modelo (los datos), las vistas (las pantallas) y las operaciones asociadas a los botones (controlador). Luego cada uno se encargaría de una parte.

10 CONTROL DE VERSIONES

La aplicación se ha realizado utilizando el control de versiones git y el servidor bitbucket.

Para clonar el proyecto debemos realizar:

git clone <https://bitbucket.org/ceeddaw/prg1314finalgrupo1>

11 PROBLEMAS ENCONTRADOS

Ha sido difícil encontrar ejemplo completos de java SE, Netbeans y IGU.

12 MEJORAS.

Hacer la ventana inicial con un menú.

13 BIBLIOGRAFIA.

1. Tutorial Programación pantalla de Acceso(Login) en JAVA Netbeans
<http://www.youtube.com/watch?v=ayCeQh8IGgA>
2. Tienda Online con Java EE
<https://code.google.com/p/tiendaonlinelpijdyb/>
3. Proyecto Universidad
<http://www.youtube.com/watch?v=C5PScZ7HN-k>
4. Página web con Iconos.
<http://icones.pro/es/>
5. Instancias de una clase:
<http://migranitodejava.blogspot.com.es/2011/05/singleton.html>
6. Mini Proyecto de Facturación en Java
<http://www.youtube.com/watch?v=NQM5M-C37Qo>
7. Login a través de Netbeans con MySQL

Proyecto final programación. prg1314finalgrupo1. Paco Aldarias. CEED.

<https://www.youtube.com/watch?v=vPvCk-E8YGQ>
prg1314finalgrupo1prg1314finalgrupo1