

ANEXO 1: Diagramas de clases y fichas CRC

Índice

1. Introducción.....	2
2. Clases.....	2
2.1 Subsistema de menú y parámetros de usuario.....	2
2.1.1 <i>Diagrama de clases y jerarquía UML.....</i>	2
2.1.2 <i>Relación de clases implicadas.....</i>	2
2.2 Subsistema de consultas de clientes y artículos.....	7
2.2.1 <i>Diagrama de clases y jerarquía UML.....</i>	7
2.2.2 <i>Relación de clases implicadas.....</i>	8
2.3 Subsistema de pedidos.....	16
2.3.1 <i>Diagrama de clases y jerarquía UML.....</i>	16
2.3.2 <i>Relación de clases implicadas.....</i>	16
2.4 Subsistema de sincronización.....	25
2.4.1 <i>Diagrama de clases y jerarquía UML.....</i>	25
2.4.2 <i>Relación de clases implicadas.....</i>	26

1. Introducción

En el presente anexo mostramos los diagramas de clases y jerarquías UML de los diferentes subsistemas. También se muestran las fichas CRC (clase, responsabilidad, colaboración).

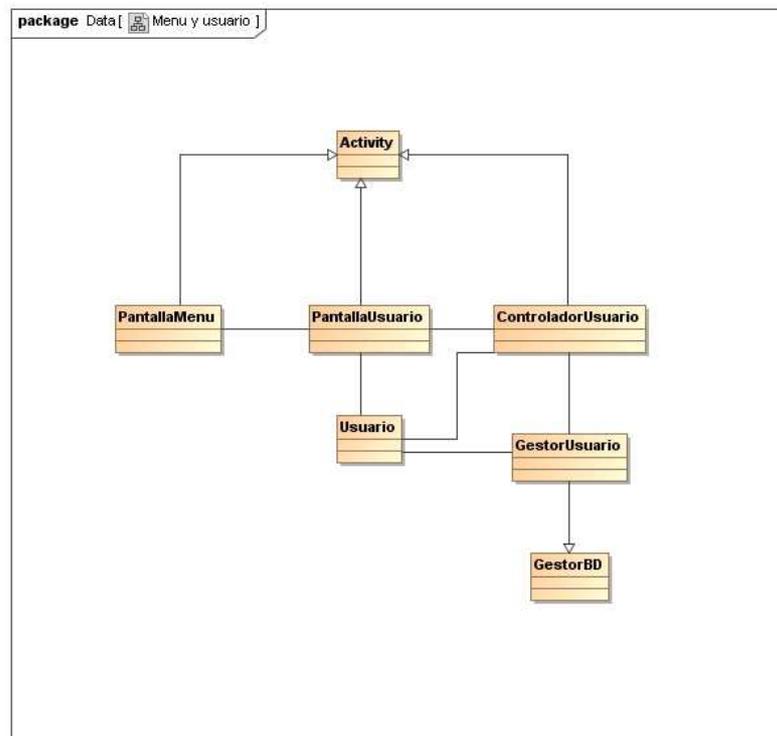
2. Clases

2.1 Subsistema de menú y parámetros de usuario

A través la pantalla del menú accederemos al resto de opciones de la aplicación. El apartado de parámetros de usuario nos permitirá configurar las particularidades del agente comercial que va a utilizar la aplicación.

2.1.1 Diagrama de clases y jerarquía UML

A continuación se muestran las clases que intervienen en este subsistema.



2.1.2 Relación de clases implicadas

Aquí mostramos las clases con detalle de sus responsabilidades, atributos y métodos más importantes.

Clase: ControladorMenu	
Descripción:	Clase dedicada a la gestión del menú
Tipo:	Controlador (Control)
Características:	Concreta

Responsabilidades:	Colaboraciones / Vínculos
Gestionar las opciones del menú	
Constructores	
+ControladorMenu();	
Atributos	Métodos
	onCreate(Bundle savedInstanceState)
	onCreateOptionsMenu(Menu menu)
	parametrosUsuario(View view)
	consultarClientes(View view)
	consultarArticulos(View view)
	pedidos(View view)
	sincronizacion(View view)

Clase: PantallaMenu

Descripción:	Presenta las opciones del menú de la aplicación
Tipo:	Pantalla (Frontera)
Características:	Concreta
Responsabilidades:	Vista de las opciones de menú.
Pantalla del menú principal de la aplicación que se lanza desde ControladorMenu con las especificaciones del fichero pantalla_menu.xml	
Constructores	
+PantallaMenu();	
Atributos	Métodos
Button <i>button_Parametros</i>	
Button <i>button_Clientes</i>	
Button <i>button_Articulos</i>	
Button <i>button_Pedidos</i>	
Button <i>button_Sincronizacion</i>	

Clase: ControladorUsuario

Descripción:	Clase dedicada a la gestión de los usuarios
Tipo:	Controlador (Control)
Características:	Concreta
Responsabilidades:	Colaboraciones / Vínculos
Gestionar el alta, modificación y consulta de usuarios.	
Constructores	
+ControladorUsuario();	
Atributos	Métodos
GestorBD <code>gestorBD</code>	<code>onCreate(Bundle savedInstanceState)</code>
EditText <code>idUsuario</code>	<code>onDestroy()</code>
EditText <code>nombreUsuario</code>	<code>getAltaUsuario()</code>
EditText <code>emailUsuario</code>	<code>guardar(View view)</code>
EditText <code>passwordUsuario</code>	<code>cancelar(View view)</code>
EditText <code>idEmpresaUsuario</code>	<code>mensajeFin(String texto)</code>
EditText <code>observacionesUsuario</code>	<code>mensajeAviso(String texto)</code>
CheckBox <code>checkMuestraPw</code>	
boolean <code>altaUsuario</code>	

Clase: PantallaUsuario

Descripción:	Presenta la pantalla con los datos del usuario que utiliza la aplicación
Tipo:	Pantalla (Frontera)
Características:	Concreta
Responsabilidades:	Vista de la pantalla de usuario.
Pantalla para ver y modificar los datos del usuario de la aplicación que se lanza desde ControladorUsuario con las especificaciones del fichero pantalla_usuario.xml	
Constructores	
+PantallaUsuario();	
Atributos	Métodos
TextView <code>labelIdUsuario</code>	
EditText <code>textIdUsuario</code>	
TextView <code>labelNombreUsuario</code>	

EditText	<code>textNombreUsuario</code>	
TextView	<code>labelEmailUsuario</code>	
EditText	<code>textEmailUsuario</code>	
TextView	<code>labelPasswordUsuario</code>	
EditText	<code>textPasswordUsuario</code>	
CheckBox	<code>checkMostrarPw</code>	
TextView	<code>labelIdEmpresaUsuario</code>	
EditText	<code>textIdEmpresaUsuario</code>	
TextView	<code>labelObservUsuario</code>	
EditText	<code>textObservUsuario</code>	
Button	<code>botUsuarioGuardar</code>	
Button	<code>botUsuarioCancelar</code>	

Clase: Usuario

Descripción:	Representa los datos básicos de un usuario
Tipo:	Entidad
Características:	Concreta
Responsabilidades:	Colaboraciones / Vínculos
Identifica a un usuario	
Constructores	
+Usuario(int id, String nombre, String password, int idEmpresa, String email, String observaciones)	
Atributos	Métodos
int id	getId()
String nombre	setId(int id)
String password	getNombre()
int idEmpresa	setNombre(String nombre)
String email	getPassword()
String observaciones	setPassword(String password)
	getIdEmpresa()
	setIdEmpresa(int idEmpresa)
	getEmail()
	setEmail(String email)
	getObservaciones()
	setObservaciones(String

	observaciones)
	toString()

Clase: GestorUsuario	
Descripción:	Clase que proporciona los métodos necesarios para realizar la persistencia de la clase Usuario.
Tipo:	Gestor
Características:	Concreta
Responsabilidades:	Colaboraciones / Vínculos
Gestionar el alta, modificación y consulta de usuarios en la BD.	
Constructores	
+GestorUsuario(SQLiteDatabase bd)	
Atributos	Métodos
SQLiteDatabase bd	insertarUsuario(Usuario usuario)
	modificarUsuario(Usuario usuario)
	primerUsuario()

Clase: GestorBD	
Descripción:	Clase que proporciona los métodos necesarios para realizar la persistencia de la clase Usuario, Artículo, Cliente, Pedido y PedidoLinea.
Tipo:	Gestor
Características:	Concreta
Responsabilidades:	Colaboraciones / Vínculos
Gestionar los accesos de la BD.	
Constructores	
+GestorBD(Context ctx)	
Atributos	Métodos
final String DATABASE_NAME	open()
	close()
final String CREATE_TABLE_USUARIOS	altaUsuario(Usuario usuario)

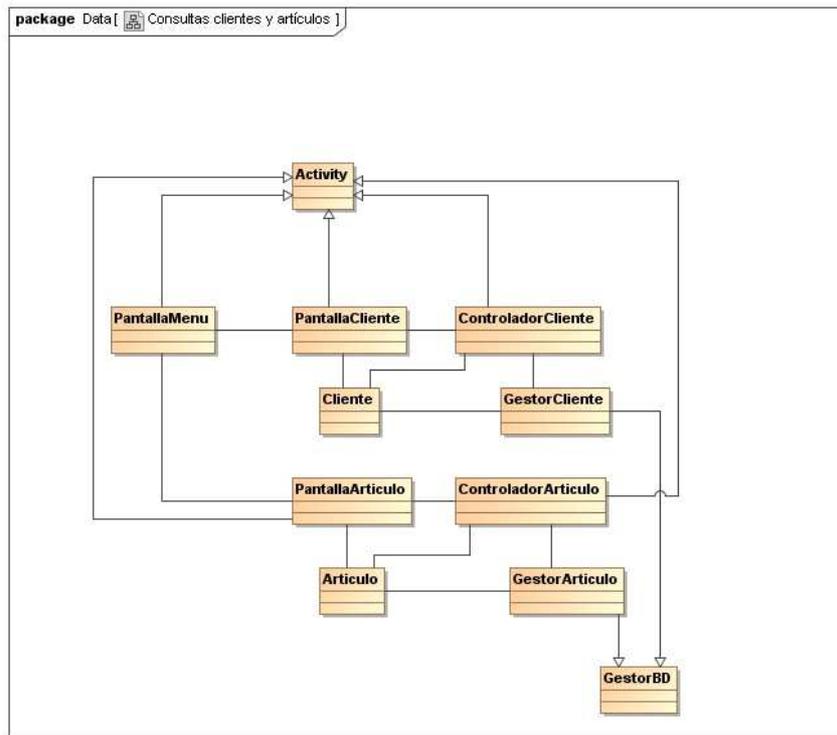
<code>final String CREATE_TABLE_CLIENTES</code>	<code>primerUsuario()</code>
<code>final String CREATE_TABLE_ARTICULOS</code>	<code>modificaUsuario(Usuario usuario)</code>
<code>final String CREATE_TABLE_PEDIDOS</code>	<code>altaCliente(Cliente cliente)</code>
<code>final String CREATE_TABLE_PEDIDOS_LINEAS</code>	<code>leeCliente(int idCliente)</code>
<code>final Context context</code>	<code>modificaCliente(Cliente cliente)</code>
<code>DatabaseHelper DBHelper</code>	<code>listaCliente(String condicion)</code>
<code>SQLiteDatabase bd</code>	<code>altaArticulo(Articulo articulo)</code>
<code>GestorUsuario gestorUsuario</code>	<code>leeArticulo(int idArticulo)</code>
<code>GestorCliente gestorCliente</code>	<code>modificaArticulo(Articulo articulo)</code>
<code>GestorArticulo gestorArticulo</code>	<code>listaArticulo(String condicion)</code>
<code>GestorPedido gestorPedido</code>	<code>altaPedido(Pedido pedido)</code>
<code>GestorPedidoLinea gestorPedidoLinea</code>	<code>leePedido(int idPedido)</code>
	<code>modificaPedido(Pedido pedido)</code>
	<code>bajaPedido(int idPedido)</code>
	<code>siguientePedido()</code>
	<code>listaPedido(String condicion)</code>
	<code>altaPedidoLinea(PedidoLinea pedidoLinea)</code>
	<code>leePedidoLinea(int idPedido, int idLinea)</code>
	<code>modificaPedidoLinea(PedidoLinea pedidoLinea)</code>
	<code>bajaPedidoLinea(int idPedido, int idLinea)</code>
	<code>siguienteLinea(int idPedido)</code>
	<code>listaPedidoLinea(int idPedido)</code>

2.2 Subsistema de consultas de clientes y artículos

Mediante este subsistema el agente comercial podrá consultar datos relativos a clientes y artículos.

2.2.1 Diagrama de clases y jerarquía UML

A continuación se muestran las clases que intervienen en este subsistema.



2.2.2 Relación de clases implicadas

Aquí mostramos las clases con detalle de sus responsabilidades, atributos y métodos más importantes.

Clase: ControladorCliente	
Descripción:	Clase dedicada a la gestión de los clientes
Tipo:	Controlador (Control)
Características:	Concreta
Responsabilidades:	Colaboraciones / Vínculos
Gestionar la consulta de clientes.	
Constructores	
+ControladorCliente();	
Atributos	Métodos
ClienteListAdapter <code>clienteListAdapter</code>	<code>onCreate(Bundle savedInstanceState)</code>
GestorBD <code>gestorBD</code>	<code>onDestroy()</code>
String <code>tipoLlamada</code>	<code>buscar(View view)</code>
int <code>idClienteRetorno</code>	<code>mensajeAviso(String texto)</code>
EditText <code>nombreClienteBuscar</code>	
EditText <code>idClienteCliente</code>	

EditText <code>nombreCliente</code>	
EditText <code>direccionCliente</code>	
EditText <code>codigo_postalCliente</code>	
EditText <code>poblacionCliente</code>	
EditText <code>provinciaCliente</code>	
EditText <code>paisCliente</code>	
EditText <code>telefonoCliente</code>	
EditText <code>movilCliente</code>	
EditText <code>emailCliente</code>	
EditText <code>forma_pagoCliente</code>	
EditText <code>observacionesCliente</code>	

Clase: PantallaCliente	
Descripción:	Presenta la pantalla con los datos del cliente
Tipo:	Pantalla (Frontera)
Características:	Concreta
Responsabilidades:	Vista de la pantalla de cliente
Pantalla para ver los datos de los clientes de la aplicación que se lanza desde ControladorCliente con las especificaciones del fichero pantalla_cliente.xml	
Constructores	
+PantallaCliente();	
Atributos	Métodos
TextView <code>lTituloClientes</code>	
TextView <code>lNombreCliente</code>	
EditText <code>nombreBuscarCliente</code>	
Button <code>botClienteBuscar</code>	
ListView <code>clientes_list</code>	
TextView <code>labNombreCliente</code>	
EditText <code>nombreCliente</code>	
EditText <code>idClienteCliente</code>	
TextView <code>labDireccionCliente</code>	
EditText <code>direccionCliente</code>	
TextView <code>lbCodigo_PostalCliente</code>	
EditText <code>codigo_PostalCliente</code>	
EditText <code>poblacionCliente</code>	

TextView <i>lbProvincia</i>	
EditText <i>provinciaCliente</i>	
EditText <i>paisCliente</i>	
TextView <i>lTelefonoCliente</i>	
EditText <i>telefonoCliente</i>	
EditText <i>movilCliente</i>	
TextView <i>lbEmailCliente</i>	
EditText <i>emailCliente</i>	
TextView <i>lbForma_PagoCliente</i>	
EditText <i>forma_pagoCliente</i>	
TextView <i>lbObservacionesCliente</i>	
EditText <i>observacionesCliente</i>	

Clase: Cliente

Descripción:	Representa los datos básicos de un cliente
Tipo:	Entidad
Características:	Concreta
Responsabilidades:	Colaboraciones / Vínculos
Identifica a un cliente	
Constructores	
+Cliente(int idCliente, String nombre, String nif, String direccion, String codigo_postal, String poblacion, String provincia, String pais, String telefono, String movil, String email, String forma_pago, String observaciones)	
Atributos	Métodos
int idCliente;	setIdCliente(int idCliente)
String nombre	getIdCliente()
String nif	setNombre(String nombre)
String direccion	String getNombre()
String codigo_postal	setNif(String nif)
String poblacion	getNif()
String provincia	setDireccion(String direccion)
String pais	getDireccion()
String telefono	setCodigo_postal(String codigo_postal)
String movil	getCodigo_postal()
String email	setPoblacion(String poblacion)

String <code>forma_pago</code>	<code>getPoblacion()</code>
String <code>observaciones</code>	<code>setProvincia(String provincia)</code>
	<code>getProvincia()</code>
	<code>setPais(String pais)</code>
	<code>getPais()</code>
	<code>setTelefono(String telefono)</code>
	<code>getTelefono()</code>
	<code>setMovil(String movil)</code>
	<code>getMovil()</code>
	<code>setEmail(String email)</code>
	<code>getEmail()</code>
	<code>setForma_pago(String forma_pago)</code>
	<code>getForma_pago()</code>
	<code>setObservaciones(String observaciones)</code>
	<code>getObservaciones()</code>
	<code>toString()</code>

Clase: GestorCliente

Descripción:	Clase que proporciona los métodos necesarios para realizar la persistencia de la clase Cliente.
Tipo:	Gestor
Características:	Concreta
Responsabilidades:	Colaboraciones / Vínculos
Gestionar el alta, modificación y consulta de clientes en la BD.	
Constructores	
+GestorCliente(SQLiteDatabase bd)	
Atributos	Métodos
SQLiteDatabase <code>bd</code>	<code>insertarCliente(Cliente cliente)</code>
	<code>modificarCliente(Cliente cliente)</code>
	<code>leeCliente(int idCliente)</code>
	<code>listaCliente(String condicion)</code>

Clase: ControladorArticulo	
Descripción:	Clase dedicada a la gestión de los artículos
Tipo:	Controlador (Control)
Características:	Concreta
Responsabilidades:	Colaboraciones / Vínculos
Gestionar la consulta de artículos.	
Constructores	
+ControladorArticulo();	
Atributos	Métodos
ArticuloListAdapter <code>articuloListAdapter</code>	<code>onCreate(Bundle savedInstanceState)</code>
GestorBD <code>gestorBD</code>	<code>onDestroy()</code>
String <code>tipoLlamada</code>	<code>buscar(View view)</code>
int <code>idArticuloRetorno</code>	<code>mensajeAviso(String texto)</code>
EditText <code>descripcionBuscarArticulo</code>	
EditText <code>idArticuloArticulo</code>	
EditText <code>descripcionArticulo</code>	
EditText <code>familiaArticulo</code>	
EditText <code>temporadaArticulo</code>	
EditText <code>precioArticulo</code>	
EditText <code>pvprArticulo</code>	
EditText <code>tieneTallasArticulo</code>	
EditText <code>talla1Articulo</code>	
EditText <code>talla2Articulo</code>	
EditText <code>talla3Articulo</code>	
EditText <code>talla4Articulo</code>	
EditText <code>talla5Articulo</code>	
EditText <code>talla6Articulo</code>	
EditText <code>talla7Articulo</code>	
EditText <code>talla8Articulo</code>	
EditText <code>talla9Articulo</code>	
EditText <code>talla10Articulo</code>	
EditText <code>observacionesArticulo</code>	
DecimalFormat <code>moneda</code>	

Clase: PantallaArticulo	
Descripción:	Presenta la pantalla con los datos del artículo
Tipo:	Pantalla (Frontera)
Características:	Concreta
Responsabilidades:	Vista de la pantalla de artículo
Pantalla para ver los datos de los artículos de la aplicación que se lanza desde ControladorArticulo con las especificaciones del fichero pantalla_articulo.xml	
Constructores	
+PantallaArticulo();	
Atributos	Métodos
TextView <i>lTituloArticulos</i>	
TextView <i>lDescripcionArticulo</i>	
EditText <i>descripcionBuscarArticulo</i>	
Button <i>botArticuloBuscar</i>	
ListView <i>articulos_list</i>	
TextView <i>labDescripcionArticulo</i>	
EditText <i>descripcionArticulo</i>	
EditText <i>idArticuloArticulo</i>	
TextView <i>labFamiliaArticulo</i>	
EditText <i>familiaArticulo</i>	
TextView <i>lbTemporadaArticulo</i>	
EditText <i>temporadaArticulo</i>	
TextView <i>lbPrecioArticulo</i>	
EditText <i>precioArticulo</i>	
TextView <i>lPvprArticulos</i>	
EditText <i>pvprArticulo</i>	
TextView <i>lbTieneTallasArticulo</i>	
EditText <i>tieneTallasArticulo</i>	
TextView <i>lbTallasArticulo</i>	
EditText <i>talla1_Articulo</i>	
EditText <i>talla2_Articulo</i>	
EditText <i>talla3_Articulo</i>	
EditText <i>talla4_Articulo</i>	
EditText <i>talla5_Articulo</i>	
EditText <i>talla6_Articulo</i>	
EditText <i>talla7_Articulo</i>	
EditText <i>talla8_Articulo</i>	

EditText <i>talla9_Articulo</i>	
EditText <i>talla10_Articulo</i>	
TextView <i>lbObservacionesArticulo</i>	
EditText <i>observacionesArticulo</i>	

Clase: Artículo	
Descripción:	Representa los datos básicos de un artículo
Tipo:	Entidad
Características:	Concreta
Responsabilidades:	Colaboraciones / Vínculos
Identifica a un artículo	
Constructores	
<pre>+Articulo(int idArticulo, String descripcion, String familia, String temporada, double precio, double pvpr, int tienetallas, String talla1, String talla2, String talla3, String talla4, String talla5, String talla6, String talla7, String talla8, String talla9, String talla10, String observaciones)</pre>	
Atributos	Métodos
<code>int idArticulo</code>	<code>setIdArticulo(int idArticulo)</code>
<code>String descripcion</code>	<code>getIdArticulo()</code>
<code>String familia</code>	<code>setDescripcion(String descripcion)</code>
<code>String temporada</code>	<code>getDescripcion()</code>
<code>double precio</code>	<code>setFamilia(String familia)</code>
<code>double pvpr</code>	<code>getFamilia()</code>
<code>int tienetallas</code>	<code>setTemporada(String temporada)</code>
<code>String talla1</code>	<code>getTemporada()</code>
<code>String talla2</code>	<code>setPrecio(double precio)</code>
<code>String talla3</code>	<code>getPrecio()</code>
<code>String talla4</code>	<code>setPvpr(double pvpr)</code>
<code>String talla5</code>	<code>getPvpr()</code>
<code>String talla6</code>	<code>setTienetallas(int tienetallas)</code>
<code>String talla7</code>	<code>getTienetallas()</code>
<code>String talla8</code>	<code>setTalla1(String talla1)</code>
<code>String talla9</code>	<code>getTalla1()</code>
<code>String talla10</code>	<code>setTalla2(String talla2)</code>
<code>String observaciones</code>	<code>getTalla2()</code>

	setTalla3(String talla3)
	getTalla3()
	setTalla4(String talla4)
	getTalla4()
	setTalla5(String talla5)
	getTalla5()
	setTalla6(String talla6)
	getTalla6()
	setTalla7(String talla7)
	getTalla7()
	setTalla8(String talla8)
	getTalla8()
	setTalla9(String talla9)
	getTalla9()
	setTalla10(String talla10)
	getTalla10()
	setObservaciones(String observaciones)
	getObservaciones()
	toString()

Clase: GestorArticulo

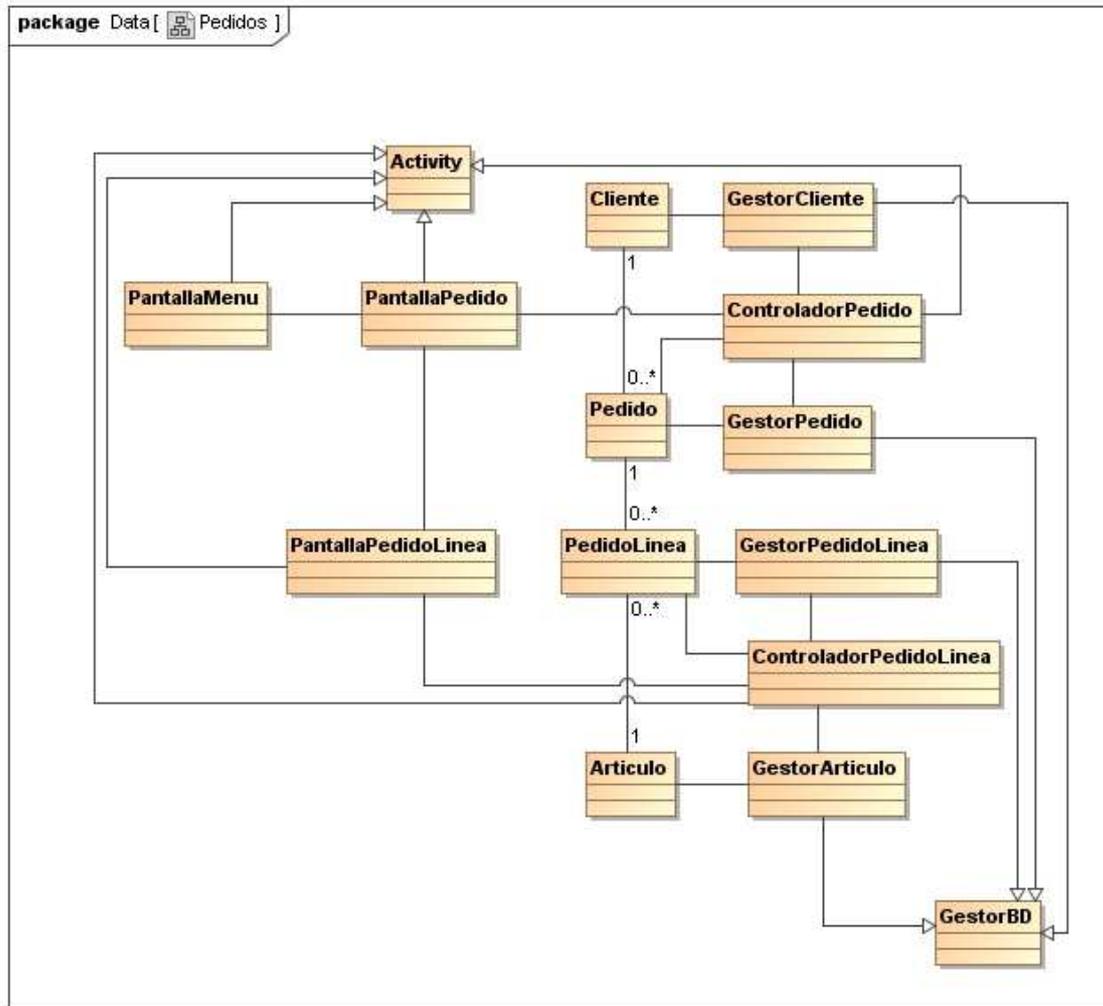
Descripción:	Clase que proporciona los métodos necesarios para realizar la persistencia de la clase Articulo.
Tipo:	Gestor
Características:	Concreta
Responsabilidades:	Colaboraciones / Vínculos
Gestionar el alta, modificación y consulta de artículos en la BD.	
Constructores	
+GestorArticulo();	
Atributos	Métodos
SQLiteDatabase bd	insertarArticulo(Articulo articulo)
	modificarArticulo(Articulo articulo)
	leeArticulo(int idArticulo)
	listaArticulo(String condicion)

2.3 Subsistema de pedidos

Este subsistema permitirá crear pedidos, modificar y consultar.

2.3.1 Diagrama de clases y jerarquía UML

A continuación se muestran las clases que intervienen en este subsistema.



2.3.2 Relación de clases implicadas

Aquí mostramos las clases con detalle de sus responsabilidades, atributos y métodos más importantes.

Clase: ControladorPedido	
Descripción:	Clase dedicada a la gestión de los pedidos
Tipo:	Controlador (Control)
Características:	Concreta
Responsabilidades:	Colaboraciones / Vínculos

Gestionar el alta, modificación y consulta de clientes.	
Constructores	
+ControladorPedido();	
Atributos	Métodos
PedidoListAdapter <code>pedidoListAdapter</code>	<code>onCreate(Bundle savedInstanceState)</code>
PedidoLineaListAdapter <code>pedidoLineaListAdapter</code>	<code>onDestroy()</code>
GestorBD <code>gestorBD</code>	<code>getAltaPedido()</code>
boolean <code>altaPedido</code>	<code>desActivaEdits()</code>
EditText <code>nombreBuscarPedido</code>	<code>activaEdits()</code>
EditText <code>idPedidoPedido</code>	<code>buscar(View view)</code>
EditText <code>idClientePedido</code>	<code>cargarListaPedidos()</code>
EditText <code>nombreClientePedido</code>	<code>cargarListaLineas()</code>
EditText <code>idFechaPedido</code>	<code>guardar(View view)</code>
EditText <code>idFechaServicio</code>	<code>cancelar(View view)</code>
EditText <code>idFechaEnviado</code>	<code>nuevoPedido(View view)</code>
EditText <code>observacionesPedido</code>	<code>bajaPedido(View view)</code>
EditText <code>cantidadPedido</code>	<code>nuevaLinea(View view)</code>
EditText <code>importePedido</code>	<code>modificaLinea(View view)</code>
EditText <code>lineaSeleccionada</code>	<code>bajaLinea(View view)</code>
EditText <code>descripcionSeleccionada</code>	<code>onActivityResult(int requestCode, int resultCode, Intent data)</code>
DecimalFormat <code>moneda</code>	<code>mensajeFin(String texto)</code>
DecimalFormat <code>unidad</code>	<code>mensajeAviso(String texto)</code>

Clase: PantallaPedido

Descripción:	Presenta la pantalla con los datos del pedido
Tipo:	Pantalla (Frontera)
Características:	Concreta
Responsabilidades:	Vista de la pantalla de pedido
Pantalla para ver los datos de los pedidos de la aplicación que se lanza desde ControladorPedido con las especificaciones del fichero pantalla_pedido.xml	
Constructores	
+PantallaPedido();	
Atributos	Métodos

TextView <i>lTituloPedidos</i>	
TextView <i>lNombreClientePedido</i>	
EditText <i>nombreBuscarPedido</i>	
Button <i>botPedidoBuscar</i>	
ListView <i>pedidos_list</i>	
Button <i>butMasPed</i>	
Button <i>butMenPed</i>	
TextView <i>labIdPedidoPedido</i>	
EditText <i>idPedidoPedido</i>	
TextView <i>labNombreClientePedido</i>	
EditText <i>idClientePedido</i>	
EditText <i>nombreClientePedido</i>	
TextView <i>labFechaPedido</i>	
EditText <i>idFechaPedido</i>	
TextView <i>labFechaServicio</i>	
EditText <i>idFechaServicio</i>	
TextView <i>labFechaEnviado</i>	
EditText <i>idFechaEnviado</i>	
TextView <i>lbObservacionesPedido</i>	
EditText <i>observacionesPedido</i>	
Button <i>botPedidoGuardar</i>	
Button <i>botPedidoCancelar</i>	
TextView <i>lLineasPedido</i>	
ListView <i>pedidosLineas_list</i>	
Button <i>butPedLinMas</i>	
Button <i>butPedLinMod</i>	
Button <i>butPedLinMenos</i>	
TextView <i>lTotalPedido</i>	
EditText <i>cantidadPedido</i>	
TextView <i>lImportePedido</i>	
EditText <i>importePedido</i>	
TextView <i>lbLineaSeleccionada</i>	
EditText <i>lineaSeleccionada</i>	
EditText <i>descripcionSeleccionada</i>	

Clase: Pedido	
Descripción:	Representa los datos básicos de un pedido
Tipo:	Entidad
Características:	Concreta
Responsabilidades:	Colaboraciones / Vínculos
Identifica a un pedido	
Constructores	
+Pedido(int idPedido, int idCliente, String fechaPedido, String fechaServicio, String fechaEnviado, String observaciones)	
Atributos	Métodos
int idPedido	setIdPedido(int idPedido)
int idCliente	getIdPedido()
String fechaPedido	setIdCliente(int idCliente)
String fechaServicio	getIdCliente()
String fechaEnviado	setFechaPedido(String fechaPedido)
String observaciones	getFechaPedido()
	setFechaServicio(String fechaServicio)
	getFechaServicio()
	setFechaEnviado(String fechaEnviado)
	getFechaEnviado()
	setObservaciones(String observaciones)
	getObservaciones()
	toString()

Clase: GestorPedido	
Descripción:	Clase que proporciona los métodos necesarios para realizar la persistencia de la clase Pedido
Tipo:	Gestor
Características:	Concreta
Responsabilidades:	Colaboraciones / Vínculos
Gestionar el alta, modificación y consulta de pedidos en la BD.	
Constructores	
+GestorPedido();	

Atributos	Métodos
SQLiteDatabase <code>bd</code>	<code>insertarPedido(Pedido pedido)</code>
	<code>modificarPedido(Pedido pedido)</code>
	<code>leePedido(int idPedido)</code>
	<code>siguientePedido()</code>
	<code>bajaPedido(int idPedido)</code>
	<code>listaPedido(String condicion)</code>

Clase: ControladorPedidoLinea

Descripción:	Clase dedicada a la gestión de las líneas de pedido
Tipo:	Controlador (Control)
Características:	Concreta
Responsabilidades:	Colaboraciones / Vínculos
Gestionar el alta, modificación y consulta de líneas de pedido	
Constructores	
+ControladorPedidoLinea();	
Atributos	Métodos
GestorBD <code>gestorBD</code>	<code>onCreate(Bundle savedInstanceState)</code>
EditText <code>idPedidoPedido</code>	<code>getAltaLinea()</code>
EditText <code>idClientePedido</code>	<code>onDestroy()</code>
EditText <code>nombreClientePedido</code>	<code>nuevoArticulo(View view)</code>
EditText <code>idFechaPedido</code>	<code>cargaArticulo()</code>
EditText <code>idFechaServicio</code>	<code>guardar(View view)</code>
EditText <code>idFechaEnviado</code>	<code>cancelar(View view)</code>
EditText <code>observacionesPedido</code>	<code>onActivityResult(int requestCode, int resultCode, Intent data)</code>
boolean <code>altaLinea</code>	<code>mensajeFin(String texto)</code>
EditText <code>lineaPedidosLineas</code>	<code>mensajeAviso(String texto)</code>
EditText <code>idArticuloArticuloPedido</code>	
EditText <code>descripcionArticuloPedido</code>	
EditText <code>familiaArticuloPedido</code>	
EditText <code>precioArticuloPedido</code>	
EditText <code>pvprArticuloPedido</code>	
EditText <code>talla1</code>	

EditText talla2	
EditText talla3	
EditText talla4	
EditText talla5	
EditText talla6	
EditText talla7	
EditText talla8	
EditText talla9	
EditText talla10	
EditText talla1ArticuloPedido	
EditText talla2ArticuloPedido	
EditText talla3ArticuloPedido	
EditText talla4ArticuloPedido	
EditText talla5ArticuloPedido	
EditText talla6ArticuloPedido	
EditText talla7ArticuloPedido	
EditText talla8ArticuloPedido	
EditText talla9ArticuloPedido	
EditText talla10ArticuloPedido	
EditText cantidadLinea	
EditText importeLinea	
EditText observacionesArticuloPedido	
double precioLinea	
DecimalFormat moneda	
DecimalFormat unidad	

Clase: PantallaPedidoLinea

Descripción:	Presenta la pantalla con los datos de una línea de pedido
Tipo:	Pantalla (Frontera)
Características:	Concreta
Responsabilidades:	Vista de la pantalla de una línea de pedido
Pantalla para ver los datos de una línea de un pedido que se lanza desde ControladorPedidoLinea con las especificaciones del fichero pantalla_pedido linea.xml	
Constructores	

+PantallaPedidoLinea();	
Atributos	Métodos
TextView <i>lTituloPedidosLineas</i>	
TextView <i>labIdPedidoPedidoLinea</i>	
EditText <i>idPedidoPedidoLinea</i>	
TextView <i>labNombreClientePedidoLinea</i>	
EditText <i>idClientePedidoLinea</i>	
EditText <i>nombreClientePedidoLinea</i>	
TextView <i>labFechaPedidoLinea</i>	
EditText <i>idFechaPedidoLinea</i>	
TextView <i>labFechaServicioLinea</i>	
EditText <i>idFechaServicioLinea</i>	
TextView <i>labFechaEnviadoLinea</i>	
EditText <i>idFechaEnviadoLinea</i>	
TextView <i>lbObservacionesPedidoLinea</i>	
EditText <i>observacionesPedidoLinea</i>	
TextView <i>lLineasPedido</i>	
EditText <i>lineaPedidosLineas</i>	
TextView <i>labDescripcionArticuloPedido</i>	
EditText <i>descripcionArticuloPedido</i>	
EditText <i>idArticuloArticuloPedido</i>	
Button <i>butArticulo</i>	
TextView <i>labFamiliaArticuloPedido</i>	
EditText <i>familiaArticuloPedido</i>	
TextView <i>lbPrecioArticuloPedido</i>	
EditText <i>precioArticuloPedido</i>	
TextView <i>lPvprArticulosPedido</i>	
EditText <i>pvprArticuloPedido</i>	
EditText <i>talla1</i>	
EditText <i>talla2</i>	
EditText <i>talla3</i>	
EditText <i>talla4</i>	
EditText <i>talla5</i>	
EditText <i>talla6</i>	
EditText <i>talla7</i>	
EditText <i>talla8</i>	
EditText <i>talla9</i>	
EditText <i>talla10</i>	
EditText <i>talla1_ArticuloPedido</i>	

EditText <i>talla2_ArticuloPedido</i>	
EditText <i>talla3_ArticuloPedido</i>	
EditText <i>talla4_ArticuloPedido</i>	
EditText <i>talla5_ArticuloPedido</i>	
EditText <i>talla6_ArticuloPedido</i>	
EditText <i>talla7_ArticuloPedido</i>	
EditText <i>talla8_ArticuloPedido</i>	
EditText <i>talla9_ArticuloPedido</i>	
EditText <i>talla10_ArticuloPedido</i>	
TextView <i>lCantidadLinea</i>	
EditText <i>cantidadLinea</i>	
TextView <i>lImporteLinea</i>	
EditText <i>importeLinea</i>	
TextView <i>lbObservacionesArticuloPedido</i>	
EditText <i>observacionesArticuloPedido</i>	
Button <i>botPedidoGuardarLinea</i>	
Button <i>botPedidoCancelarLinea</i>	

Clase: PedidoLinea

Descripción:	Representa los datos básicos de una línea de pedido
Tipo:	Entidad
Características:	Concreta
Responsabilidades:	Colaboraciones / Vínculos
Identifica a una línea de pedido	
Constructores	
<pre>+PedidoLinea(int idPedido, int idLinea, int idArticulo, int canTalla1, int canTalla2, int canTalla3, int canTalla4, int canTalla5, int canTalla6, int canTalla7, int canTalla8, int canTalla9, int canTalla10, int cantidad, String observaciones)</pre>	
Atributos	Métodos
<code>int idPedido</code>	<code>setIdPedido(int idPedido)</code>
<code>int idLinea</code>	<code>getIdPedido()</code>
<code>int idArticulo</code>	<code>setIdLinea(int idLinea)</code>
<code>int canTalla1</code>	<code>getIdLinea()</code>
<code>int canTalla2</code>	<code>setIdArticulo(int idArticulo)</code>
<code>int canTalla3</code>	<code>getIdArticulo()</code>

<code>int canTalla4</code>	<code>setCanTalla1(int canTalla1)</code>
<code>int canTalla5</code>	<code>getCanTalla1()</code>
<code>int canTalla6</code>	<code>setCanTalla2(int canTalla2)</code>
<code>int canTalla7</code>	<code>getCanTalla2()</code>
<code>int canTalla8</code>	<code>setCanTalla3(int canTalla3)</code>
<code>int canTalla9</code>	<code>getCanTalla3()</code>
<code>int canTalla10</code>	<code>setCanTalla4(int canTalla4)</code>
<code>int cantidad</code>	<code>getCanTalla4()</code>
<code>String observaciones</code>	<code>setCanTalla5(int canTalla5)</code>
	<code>getCanTalla5()</code>
	<code>setCanTalla6(int canTalla6)</code>
	<code>getCanTalla6()</code>
	<code>setCanTalla7(int canTalla7)</code>
	<code>getCanTalla7()</code>
	<code>setCanTalla8(int canTalla8)</code>
	<code>getCanTalla8()</code>
	<code>setCanTalla9(int canTalla9)</code>
	<code>getCanTalla9()</code>
	<code>setCanTalla10(int canTalla10)</code>
	<code>getCanTalla10()</code>
	<code>setCantidad(int cantidad)</code>
	<code>getCantidad()</code>
	<code>setTotalCantidadTallas()</code>
	<code>setObservaciones(String observaciones)</code>
	<code>getObservaciones()</code>
	<code>toString()</code>

Clase: GestorPedidoLinea

Descripción:	Clase que proporciona los métodos necesarios para realizar la persistencia de la clase PedidoLinea.
Tipo:	Gestor
Características:	Concreta
Responsabilidades:	Colaboraciones / Vínculos
Gestionar el alta, modificación y consulta de líneas de pedido en la BD.	
Constructores	
+GestorPedidoLinea();	

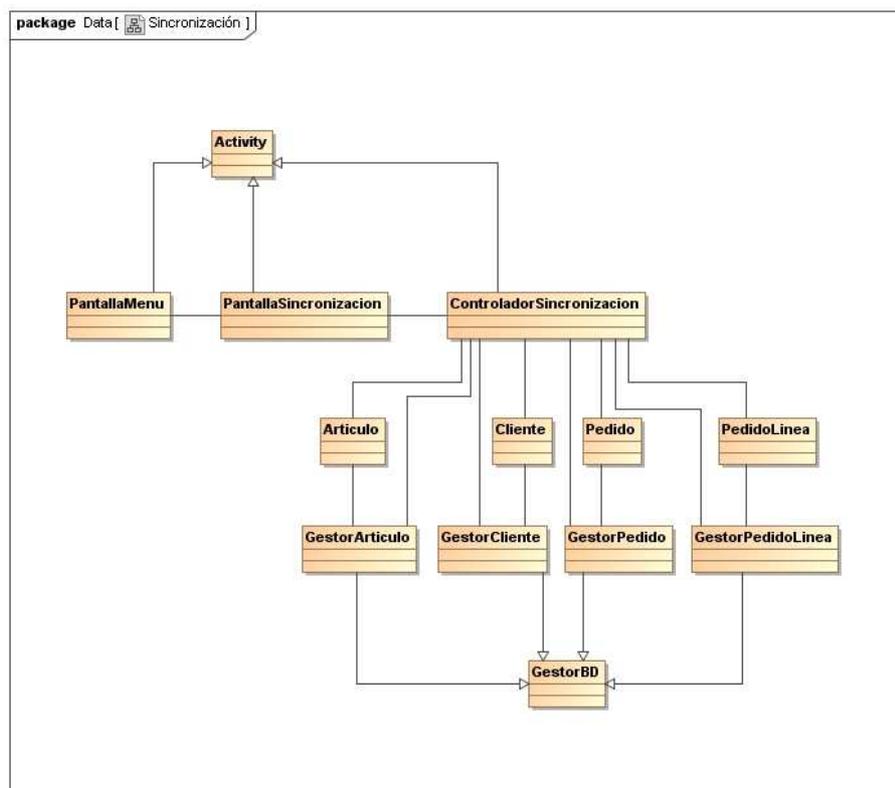
Atributos	Métodos
SQLiteDatabase <code>bd</code>	<code>insertarPedidoLinea(PedidoLinea pedidoLinea)</code>
	<code>modificarPedidoLinea(PedidoLinea pedidoLinea)</code>
	<code>bajaPedidoLinea(int idPedido, int idLinea)</code>
	<code>bajaPedido(int idPedido)</code>
	<code>leePedidoLinea(int idPedido, int idLinea)</code>
	<code>siguienteLinea(int idPedido)</code>
	<code>listaPedidoLinea(int idPedido)</code>

2.4 Subsistema de sincronización

Mediante este subsistema se intercambiará la información entre la aplicación y el ERP de la central. Por un lado, recibirá las modificaciones de clientes y artículos de la central, y por otro, enviará los nuevos pedidos que se han creado.

2.4.1 Diagrama de clases y jerarquía UML

A continuación se muestran las clases que intervienen en este subsistema.



2.4.2 Relación de clases implicadas

Aquí mostramos las clases con detalle de sus responsabilidades, atributos y métodos más importantes.

Clase: ControladorSincronizacion	
Descripción:	Clase dedicada a la gestión de la sincronización
Tipo:	Controlador (Control)
Características:	Concreta
Responsabilidades:	Colaboraciones / Vínculos
Gestionar la sincronización entre la aplicación y la central.	
Constructores	
+ControladorSincronizacion();	
Atributos	Métodos
GestorBD <code>gestorBD</code>	<code>onCreate(Bundle savedInstanceState)</code>
final static String <code>path</code>	<code>onDestroy()</code>
final static String <code>RECIBIR</code>	<code>sincronizar(View view)</code>
final static String <code>ENVIAR</code>	<code>mensajeAviso(String texto)</code>
final static String <code>TIPO</code>	
final static String <code>EOL</code>	
final static String <code>TOKEN</code>	
EditText <code>logRecibirDatos</code>	
EditText <code>logEnviarDatos</code>	
Button <code>butSincronizar</code>	
ProgressBar <code>pbClientes</code>	
ProgressBar <code>pbArticulos</code>	
ProgressBar <code>pbPedidos</code>	
ProgressBar <code>pbLineas</code>	

Clase: PantallaSincronizacion	
Descripción:	Presenta la pantalla con la sincronización entre la aplicación y la central
Tipo:	Pantalla (Frontera)

Características:	Concreta
Responsabilidades:	Vista de la pantalla de sincronización
Pantalla que muestra la sincronización entre la aplicación y la central. Se lanza desde ControladorSincronizacion con las especificaciones del fichero pantalla_sincronizacion.xml	
Constructores	
+PantallaSincronizacion();	
Atributos	Métodos
TextView <i>lTituloSincronizacion</i>	
TextView <i>lTituloRecibirDatos</i>	
TextView <i>lRecibirClientes</i>	
ProgressBar <i>pbClientes</i>	
TextView <i>lRecibirArticulos</i>	
ProgressBar <i>pbArticulos</i>	
TextView <i>lRecibirLog</i>	
EditText <i>logRecibirDatos</i>	
TextView <i>lTituloEnviarDatos</i>	
TextView <i>lEnviarPedidos</i>	
ProgressBar <i>pbPedidos</i>	
TextView <i>lEnviarLineas</i>	
ProgressBar <i>pbLineas</i>	
TextView <i>lEnviarLog</i>	
EditText <i>logEnviarDatos</i>	
Button <i>butSincronizar</i>	