

PEDIDOS

Vamos a desarrollar una aplicación web para realizar pedidos, similar a una tienda web.

Definiremos, para ello, las fases de análisis, diseño e implementación:

- En el análisis definiremos:
 - La funcionalidad de la aplicación y sus limitaciones.
 - Detallaremos los datos que queremos almacenar.
 - Desarrollaremos el modelo ER para la BD.
- En el diseño se definen:
 - Las pantallas que verá el usuario.
 - Los ficheros que formarán parte de la aplicación y como se pasarán los parámetros entre ellos.
 - La estructura de datos para el carrito de la compra y como manipularla.
 - La base de datos.
- En la implementación: se escriben los ficheros de la aplicación.

La aplicación es básicamente una tienda web sencilla. Se espera que tenga la funcionalidad habitual de una tienda Online. La única diferencia es que, como los restaurantes son de la misma empresa, el pedido no requiere pago. Tampoco es necesario especificar la dirección de envío.

Análisis de requisitos

Aplicación para el Departamento de Pedidos para una cadena de restaurantes. Los restaurantes de la cadena utilizarán la aplicación web para realizar pedidos de comida, bebida y materiales.

La aplicación debe permitir:

- Consultar las categorías.
- Consultar los productos.
- Añadir una o más unidades de un producto al pedido.
- Consultar el pedido del carrito y eliminar productos de este.
- Realizar el pedido, introduciéndolo en la BD y enviando correos de confirmación al restaurante que hace el pedido y al Departamento de Pedidos de la empresa.

Para entrar en la aplicación es necesario autenticarse. Se supone que en cada restaurante hay un responsable de pedidos con su usuario y clave para acceder a la aplicación.

De cada categoría se quiere almacenar su código, su nombre y su descripción. De los productos, su código, nombre, descripción, peso, cantidad en stock y categoría a la que pertenecen. Cada producto pertenece a una categoría.

De cada pedido interesa saber:

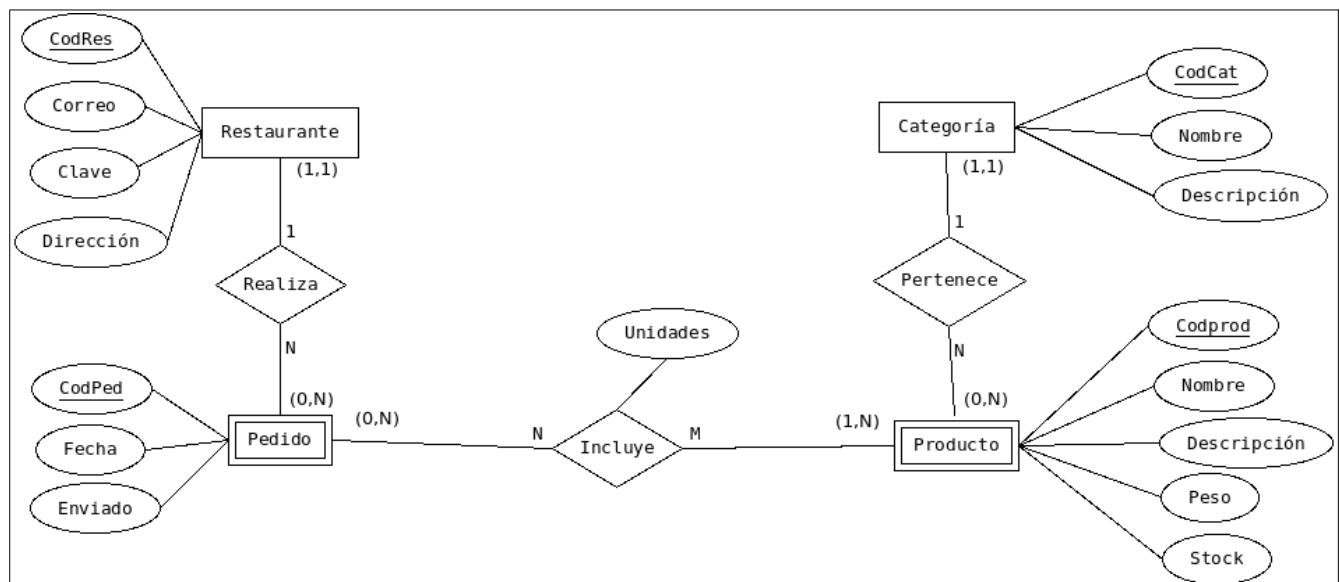
1. El restaurante que lo realizó.
2. Los productos que se pidieron, incluyendo la cantidad de unidades de cada producto.
3. Si ha sido enviado ya o no.
4. La fecha en la que se realizó el pedido.

Los pedidos se introducen en la base de datos como no enviados. Cuando se envíen el departamento de pedidos los marcará como enviados (directamente en la base de datos, la aplicación no se ocupa de esto).

De los restaurantes se guarda la siguiente información:

1. El código.
2. El correo electrónico. El correo es el nombre de usuario para acceder a la aplicación.
3. La clave.
4. País, dirección y código postal.

Esquema entidad-relación



Limitaciones de la aplicación

- No hay panel de administración. Los usuarios, categorías y productos se tienen que introducir directamente en la base de datos.
- No hay posibilidad de auto-registro.
- No se controla el stock. Si al realizar un pedido de un producto el stock se queda negativo, el pedido se tramita igualmente.

Diseño de la aplicación

Los elementos más importantes de la aplicación son:

- a) La base de datos.
- b) El flujo de pantallas para realizar el pedido.
- c) La estructura de datos del carrito de la compra.
- d) Los ficheros que forman la aplicación y como se pasan parámetros entre ellos.
- e) El control de acceso.

Diseño lógico de la base de datos

Partiendo del modelo ER se obtienen las tablas:

- Obtendremos una tabla por cada entidad.
- Las relaciones *realiza* y *pertenece a* implican un intercambio de claves. La tabla *producto* recibirá la clave de *categoría* como *clave ajena*. La tabla *pedido* recibirá la clave de *restaurante* como *clave ajena*.
- Por otro lado, la relación *incluye* es N:M y genera una tabla con la claves de *pedido* y *producto* como *claves ajenas* y los atributos de la relación. La clave principal será compuesta por las dos claves recibidas o bien puede ser una clave con auto-incremento.

Obtenemos pues, las siguientes tablas (ya normalizadas):

Restaurantes(CodRes, Correo, Clave, País, CP, Ciudad, Dirección)

PK: {CodRes} **

CALT: UNIQUE(Correo)

Pedidos(CodPed, Fecha, Enviado, Restaurante)

PK: {CodPed} **

FK: {Restaurante → Restaurantes(CodRes)} *

RE: NOTNULL(Restaurante)

RI: NOTNULL(Fecha) → Restricción de usuario: es importante que un pedido tenga fecha

RI: NOTNULL(Enviado) DEFAULT 0

Categorías(CodCat, Nombre, Descripción)

PK: {CodCat} **

CALT: UNIQUE(Nombre)

RI: NOTNULL(Nombre) → RI integridad de usuario: es importante que todas las categorías tengan nombre

Productos(CodProd, Nombre, Descripción, Peso, Stock, Categoría)

PK: {CodProd} **

FK: {Categoría → Categorías(CodCat)} *

RE: NOTNULL(Categoría)

Pedidos-Productos(CodPedProd, Pedido, Producto, Unidades)

PK: {CodPedProd} **

FK: {Pedido → Pedidos(CodPed), Producto → Productos(CodPro)} *

RI: NOTNULL(Unidades) → para que tenga sentido (Restricción de usuario)

RE: NOTNULL(Pedido) , NOTNULL(Producto) → todo pedido debe tener al menos un producto

CALT: UNIQUE({Pedido, Producto})

*Faltan las RI de las claves ajenas, en principio: Delete → restrict, Update → restrict

**Los códigos son enteros con auto-incremento

Diseño físico de la base de datos

A continuación el script que genera la BD:

```
-- phpMyAdmin SQL Dump
-- version 4.7.0
-- https://www.phpmyadmin.net/
--
-- Servidor: 127.0.0.1
-- Tiempo de generación: 16-10-2017 a las 20:23:04
-- Versión del servidor: 10.1.25-MariaDB
-- Versión de PHP: 5.6.31

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Base de datos: `pedidos`
--

--
-- Estructura de tabla para la tabla `categoria`
--

CREATE TABLE `categoria` (
  `CodCat` int(11) NOT NULL,
  `Nombre` varchar(45) NOT NULL,
  `Descripcion` varchar(200) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

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

INSERT INTO `categoria` (`CodCat`, `Nombre`, `Descripcion`) VALUES
(1, 'Comida', 'Platos e ingredientes'),
(2, 'Bebidas sin', 'Bebidas sin alcohol'),
(3, 'Bebidas con', 'Bebidas con alcohol');
```

```
--
-- Estructura de tabla para la tabla `pedidos`
--
```

```
CREATE TABLE `pedidos` (
  `CodPed` int(11) NOT NULL,
  `Fecha` datetime NOT NULL,
  `Enviado` int(11) NOT NULL DEFAULT 0, -- No enviado
  `Restaurante` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--
-- Estructura de tabla para la tabla `pedidosproductos`
--
```

```
CREATE TABLE `pedidosproductos` (
  `CodPredProd` int(11) NOT NULL,
  `CodPed` int(11) NOT NULL,
  `CodProd` int(11) NOT NULL,
  `Unidades` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--
-- Estructura de tabla para la tabla `productos`
--
```

```
CREATE TABLE `productos` (
  `CodProd` int(11) NOT NULL,
  `Nombre` varchar(45) DEFAULT NULL,
  `Descripcion` varchar(90) NOT NULL,
  `Peso` float NOT NULL,
  `Stock` int(11) NOT NULL,
  `CodCat` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

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

```

INSERT INTO `productos` (`CodProd`, `Nombre`, `Descripcion`, `Peso`, `Stock`, `CodCat`)
VALUES
(1, 'Harina', '8 paquetes de 1kg de harina cada uno', 8, 100, 1),
(2, 'Azúcar', '20 paquetes de 1kg cada uno', 20, 3, 1),
(3, 'Agua 0.5', '100 botellas de 0.5 litros cada una', 51, 100, 2),
(4, 'Agua 1.5', '20 botellas de 1.5 litros cada una', 31, 50, 2),
(5, 'Cerveza Alhambra tercio', '24 botellas de 33cl', 10, 0, 3),
(6, 'Vino tinto Rioja 0.75', '6 botellas de 0.75 ', 5.5, 10, 3);

-- -----
--
-- Estructura de tabla para la tabla `restaurantes`
--

CREATE TABLE `restaurantes` (
  `CodRes` int(11) NOT NULL,
  `Correo` varchar(90) NOT NULL,
  `Clave` varchar(45) NOT NULL,
  `Pais` varchar(45) NOT NULL,
  `CP` int(5) DEFAULT NULL,
  `Ciudad` varchar(45) NOT NULL,
  `Direccion` varchar(200) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

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

INSERT INTO `restaurantes` (`CodRes`, `Correo`, `Clave`, `Pais`, `CP`, `Ciudad`,
`Direccion`) VALUES
(1, 'madrid1@empresa.com', '1234', 'España', 28002, 'Madrid', 'C/ Padre Claret, 8'),
(2, 'cadiz1@empresa.com', '1234', 'España', 11001, 'Cádiz', 'C/ Portales, 2 ');

--
-- Índices para tablas volcadas
--

--
-- Indices de la tabla `categoria`
--
ALTER TABLE `categoria`
  ADD PRIMARY KEY (`CodCat`),
  ADD UNIQUE KEY `UN_NOM_CAT` (`Nombre`);

--
-- Indices de la tabla `pedidos`
--
ALTER TABLE `pedidos`
  ADD PRIMARY KEY (`CodPed`),
  ADD KEY `Restaurante` (`Restaurante`);

--
-- Indices de la tabla `pedidosproductos`
--

```

```

ALTER TABLE `pedidosproductos`
  ADD PRIMARY KEY (`CodPredProd`),
  ADD KEY `CodPed` (`CodPed`),
  ADD KEY `CodProd` (`CodProd`);

--
-- Indices de la tabla `productos`
--
ALTER TABLE `productos`
  ADD PRIMARY KEY (`CodProd`);

--
-- Indices de la tabla `restaurantes`
--
ALTER TABLE `restaurantes`
  ADD PRIMARY KEY (`CodRes`),
  ADD UNIQUE KEY `UN_RES_COR` (`Correo`);

--
-- AUTO_INCREMENT de las tablas volcadas
--
--
-- AUTO_INCREMENT de la tabla `categoria`
--
ALTER TABLE `categoria`
  MODIFY `CodCat` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;
--
-- AUTO_INCREMENT de la tabla `pedidos`
--
ALTER TABLE `pedidos`
  MODIFY `CodPed` int(11) NOT NULL AUTO_INCREMENT;
--
-- AUTO_INCREMENT de la tabla `pedidosproductos`
--
ALTER TABLE `pedidosproductos`
  MODIFY `CodPredProd` int(11) NOT NULL AUTO_INCREMENT;
--
-- AUTO_INCREMENT de la tabla `productos`
--
ALTER TABLE `productos`
  MODIFY `CodProd` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=7;
--
-- AUTO_INCREMENT de la tabla `restaurantes`
--
ALTER TABLE `restaurantes`
  MODIFY `CodRes` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;
--
-- Restricciones para tablas volcadas
--
--
--
-- Filtros para la tabla `productos`
--
ALTER TABLE `productos` -- Falta

```

```

    ADD CONSTRAINT `productos_ibfk_1` FOREIGN KEY (`CodCat`) REFERENCES `categorias`
    (`CodCat`);

--
-- Filtros para la tabla `pedidos`
--
ALTER TABLE `pedidos`
  ADD CONSTRAINT `pedidos_ibfk_1` FOREIGN KEY (`Restaurante`) REFERENCES `restaurantes`
  (`CodRes`);

--
-- Filtros para la tabla `pedidosproductos`
--
ALTER TABLE `pedidosproductos`
  ADD CONSTRAINT `pedidosproductos_ibfk_1` FOREIGN KEY (`CodPed`) REFERENCES `pedidos`
  (`CodPed`),
  ADD CONSTRAINT `pedidosproductos_ibfk_2` FOREIGN KEY (`CodProd`) REFERENCES `productos`
  (`CodProd`);
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```

Para insertar un pedido en la BD hay que insertar una fila en la tabla de *pedidos* y una en la tabla de *pedidosproductos* por cada producto diferente que incluya el pedido.

Inserción de un pedido

Tabla pedidos:

CodPed	Fecha	Enviado	Restaurante
1000	01/09/2019	0	1

Productos de dicho pedido → Tabla pedidosproductos:

CodPedProd	CodPed	CodProd	Unidades
20001	1000	16	100
20002	1000	20	150

Diagrama de flujo de pantallas

Vamos a representarlas pantallas por las que pasa el usuario al realizar una operación. Los cuadrados representan pantallas y las flechas acciones que llevan de unas a otras.

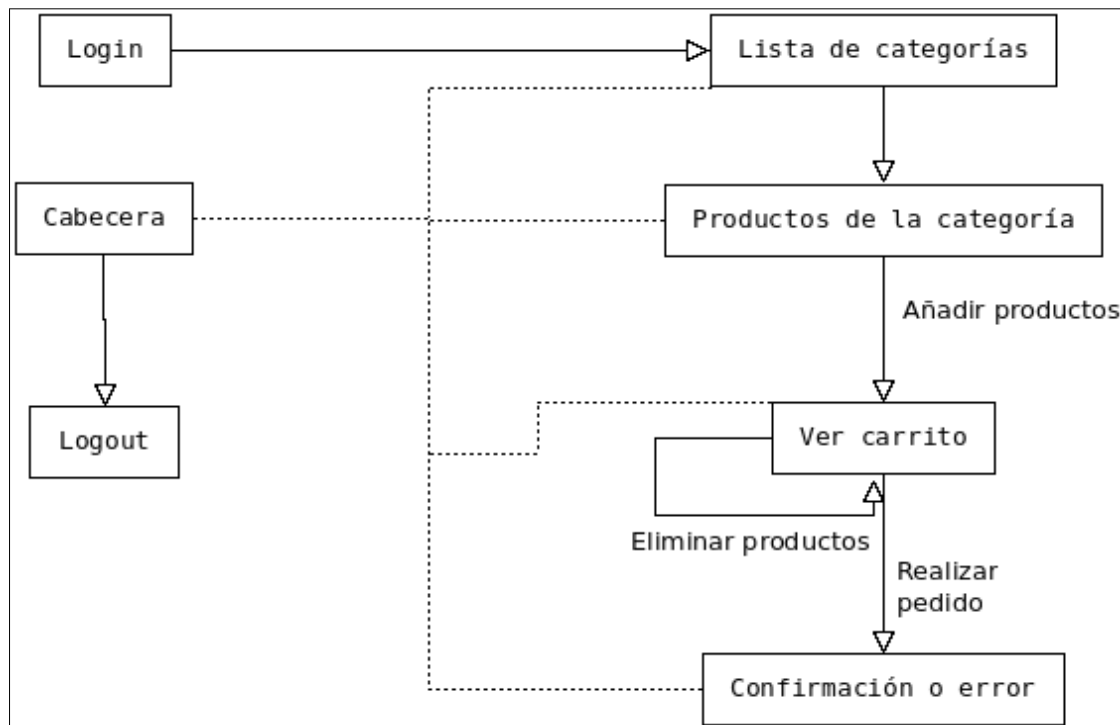
El punto de entrada a la aplicación es el *login* donde el usuario introduce un nombre y contraseña válidos para poder acceder a la aplicación.

Tras hacer el *login* con éxito se redirige al usuario a la página principal, que muestra las categorías existentes. Al seleccionar una categoría, se accede a sus productos. Tanto categorías como productos se leen de la BD (ya que pueden cambiar dinámicamente). En “productos de la categoría” se muestran los datos de los productos de cierta categoría y se permite añadir un número variable de unidades al pedido. Si se añade algún producto, se redirige al usuario a “ver carrito”.

En “ver carrito” se muestra en detalle el estado del pedido, se ofrece la posibilidad de eliminar productos y se puede confirmar el pedido. Al confirmar el pedido se muestra un mensaje de confirmación o error, según el caso.

En todas las pantallas (salvo *login* y *logout*) habrá una cabecera con el nombre del restaurante y vínculo para:

- Ver carrito
- Lista de categorías
- Cerrar la sesión



El carrito de la compra

La estructura de datos para el carrito de la compra es uno de los puntos críticos de la aplicación. Para almacenarlo se utilizará una variable de sesión.

El carrito será un *array* asociativo donde las claves representan el código de un producto y el valor el número de unidades pedidas.

El *array* comienza vacío. Cuando se añade un producto al pedido, se comprueba si ya hay en el *array* algún elemento que tenga como clave el código del producto (actuando en consecuencia). Si no lo hay se añade el código del producto como clave y el número de unidades como valor.

Ejemplo: carrito vacío y se añaden 2 unidades del producto con código 4:

`$_SESSION['carrito'][$codigo] = 2`

`$_SESSION['carrito']` →

4
2

Ahora se añaden 3 unidades del producto 5:

`$_SESSION['carrito']` →

4	5
2	3

Se añaden 5 unidades del producto de 4:

`$_SESSION['carrito']` →

4	5
7	3

Se eliminan 3 unidades del producto 4:

`$_SESSION['carrito']` →

4	5
4	3

Se eliminan 4 unidades del producto 4:

`$_SESSION['carrito']` →

5
3

Control de acceso

Al realizar un *login* con éxito se crea una nueva sesión y dos variables de sesión:

- Un *array* con dos campos: uno para guardar el nombre del usuario (el correo del restaurante) y otro para su código (del restaurante) así ya no hay que buscarlo en la BD mas adelante.
- La variable para el carrito de la compra.

El resto de los ficheros de la aplicación comienza uniéndose a la sesión y comprobando que la primera de estas variables existe. Si no se han creado es que el usuario no ha hecho *login* y, por tanto, no puede acceder. En este caso se redirige a la página de *login*.

Ficheros de la aplicación

Resumen de qué ficheros forman parte de la aplicación y como se comunican entre ellos.

Ruta	Descripción	Parámetros	Redirige a
login.php	Formulario de login	<code>\$_GET['redirigido']</code> <code>\$_POST['usuario']</code> <code>\$_POST['clave']</code>	login.php?error=TRUE categorias.php
logout.php	Cierra la sesión		
sesiones.php	Comprueba que el usuario haya iniciado sesión correctamente.		login.php (si no ha iniciado sesión)
categorias.php	Muestra la lista de categorías con vínculos a productos.php?categorias=codigo		

cabecera.php	Cabecera con vínculos para ver el carrito, las categorías o cerrar sesión		
productos.php	Muestra los productos de la categoría, permite añadir al carro de la compra	\$_GET['categoria'], el código de la categoría	
carrito.php	Muestra el carro de la compra, permite quitar productos y confirmar el pedido.		
añadir.php	Añade productos al carrito	\$_POST['cod'] \$_POST['unidades']	carrito.php
eliminar.php	Elimina productos del carrito	\$_POST['cod'] \$_POST['unidades']	carrito.php
procesarPedido.php	Inserta el pedido en la base de datos, envía correos de confirmación y muestra mensajes de error o éxito.		
bd.php	Librería de funciones de la BD		
correo.php	Librería de funciones para enviar correo		

Implementación

Se ve en clase ...