



Base de Datos II: Segunda Práctica

Nombre: Daniel Salanova Dmitriyev

DNI: 49610682G

Grupo: Oracle

Asignatura: 21726 - Base de dades II

Profesor: Jaume Jaume Mayol

Índice

Índice	1
1. Introducción	2
2. Funcionalidades	2
3. Suposiciones	3
4. Exclusiones	3
5. Modelo	4
6. Funciones, procedures y eventos	9
6.1 Procedures	10
6.2 Eventos	10
6.3 Triggers	10
7. Vistas	11
8. Aplicación	11
8.1 Home.php	11
8.2 Details.php	12
8.3 Checkout.php	12
8.4 Register.php	12
8.5 Login.php	13
9. Conclusiones	13

1. Introducción

Para esta práctica lo que se pretende crear es una versión simplificada del gran conocido Amazon. A grandes rasgos se deberá gestionar completamente la consulta del catálogo de compra e implementar las funcionalidades correspondientes a la compra.

A lo largo del documento se explicará la solución aportada y se harán incisos para detallar las suposiciones efectuadas, del hecho de trabajar como único integrante del grupo, y las funcionalidades desarrolladas.

2. Funcionalidades

El hecho de hacerlo con más o menos integrantes condicionarán en gran medida las principales funcionalidades de la página web, pero las podemos clasificar en 2 grupos diferentes:

- Funcionalidades referentes a consulta del catálogo:
 - Búsqueda por categorías: De una lista de categorías, al clicar encima de una se filtrarán los productos.
 - Barra buscadora: Filtrar productos por nombre o conjunto de caracteres.
 - Detalles del producto: Al pulsar encima de un producto nos redirigirá a una nueva pantalla con toda la información referente al producto, productos similares y comentarios.
- Funcionalidades de usuarios:
 - Los usuarios pueden crear cuentas en la página web vía un formulario, indicando el tipo de usuario que serán.
 - Los usuarios registrados podrán iniciar sesión permitiéndoles en el caso de ser compradores, llevar a cabo compras.
 - Para los usuarios compradores, podrán añadir a un carrito un conjunto de productos seleccionados para su futura compra.
 - Los usuarios compradores que hayan hecho una compra de un producto, y no hayan comentado nada de él tendrán abierto un formulario de comentario en los detalles del producto correspondiente.
 - A la hora de comprar, el usuario comprador podrá escoger entre los domicilios que tenga, para indicar la dirección de envío.
 - Los productos pueden encontrarse en oferta o no, lo que incluye descuentos en caso de estar de oferta.

3. Suposiciones

Supondremos que tendremos un listado completo de países, ciudades y calles para usarlos para indicar rutas de domicilios. Al mismo tiempo que datos estáticos como pueden ser zonas geográficas y distribuidoras.

Supondremos que a la hora de hacerse un pedido, y asignarse una distribuidora, se asignará manualmente un repartidor. Esta gestión no se llevará a cabo en la propia página web.

Por otro lado, se supondrá que un controlador, al ver que un producto ha tardado 5 días o más en llegar al almacén, mandará un mensaje al vendedor de dicho producto comprado mediante un formulario.

Supondremos también que los productos pueden tener 1 o más categorías. Esto provocará que surja más adelante una tabla adicional en el modelo entidad-relación.

Para simplificar la manera en la que interactúa la página web con la base de datos a la hora de registrar un usuario e indicar su domicilio, se ha incorporado la tabla *Domicilio* como tabla intermedia entre *Calle* y *Comprador*. De esta manera la creación de un domicilio simplemente dependerá de la selección de país, ciudad y calles ya predeterminadas (hay registros dentro de la base de datos) y de los datos insertados dentro del formulario como pueden ser número de puerta, bloque y letra.

Adicionalmente, supondremos que solo tenemos un método de pago existente, efectuado con tarjeta bancaria, lo que implica que no necesitamos una tabla de tipos de pago.

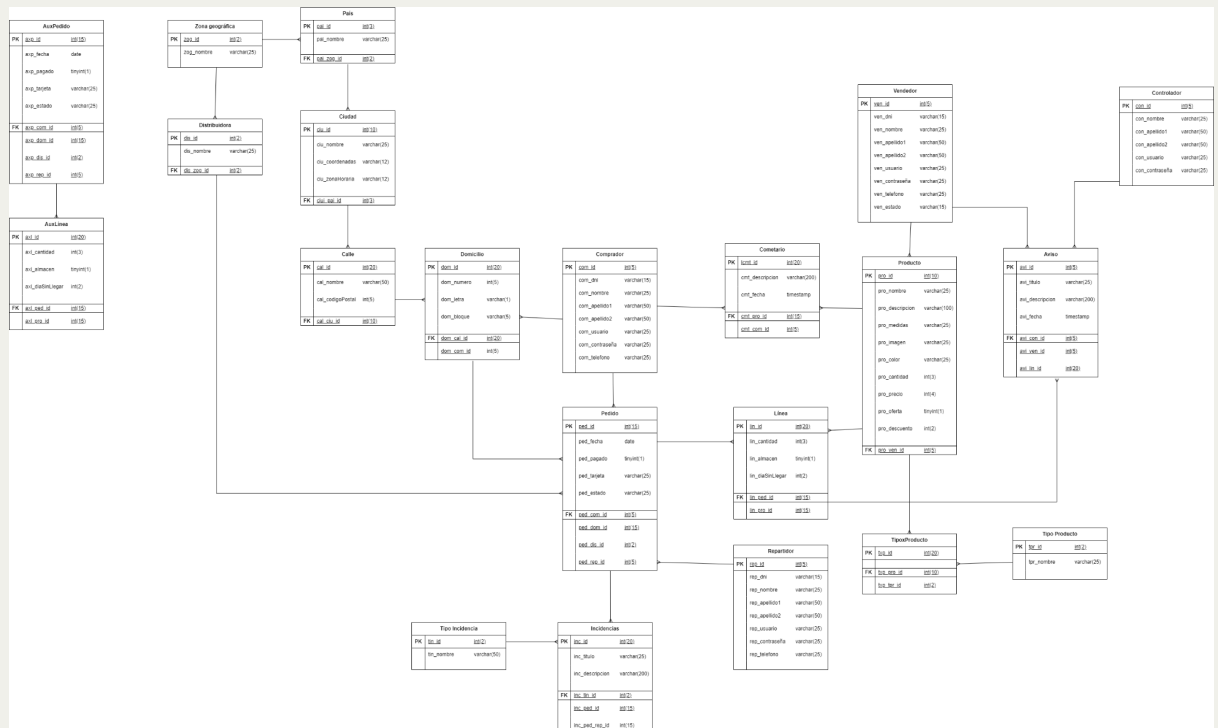
4. Exclusiones

Debido a que el proyecto es individual, parte del contenido que se excluye es el siguiente:

- Vista web de líneas que llevan 5 días sin llegar
- Mantenimiento de domicilios de un comprador (páginas de inserción, eliminación y modificación)
- Mantenimiento completo de vendedor y de los productos que pueden crear
- Mantenimiento del seguimiento del pedido y reparto.

5. Modelo

El modelo resultante del proyecto sería el siguiente:



Esta misma imagen ha sido adjuntada con la entrega, por si hubiera cualquier problema con la visualización. Ahora se procederá a explicar cada una de las diferentes tablas creadas para dar contexto, uso de cada una de ellas y su respectiva utilidad.

Tabla: Zona Geográfica

Atributos: zog_id, zog_nombre.

Tabla que se usará para almacenar diferentes continentes que tendremos en cuenta tanto para las localizaciones como para los distribuidores. Tendremos como atributos importantes su nombre, y un identificador autoincrementable.

Tabla: Distribuidora

Atributos: dis_id, dis_nombre, dis_zog_id.

En dicha clase se almacenarán todas las distribuidoras a tener presente para el envío de los pedidos. Estas distribuidoras pertenecerán a una zona en concreto y serán las encargadas de llevar a cabo el envío.

Tabla: AuxPedido

Atributos: `axp_id`, `axp_fecha`, `axp_tarjeta`, `axp_estado`, `axp_com_id`, `axp_dom_id`, `axp_dis_id`, `axp_rep_id`.

Se usará para hacer una copia de seguridad incremental de la tabla *Pedido*. Esta tabla se gestionará únicamente con el procedure *copiaSeguridad* explicado en el apartado siguiente apartado. Esta tabla es una copia exacta en cuanto a atributos de la tabla *Pedido*. En este caso el identificador no será autoincrementable ya que daría problemas a la larga a la hora de borrar o insertar nuevos registros.

Tabla: AuxLinea

Atributos: `axl_id`, `axl_cantidad`, `axl_almacen`, `axl_diaSinLlegar`, `axl_ped_id`, `axl_pro_id`.

De la misma manera que *AuxPedido*, esta tabla tendrá como única responsabilidad, guardar los datos de la copia incremental diaria de las línea de pedidos insertados este mismo día. Estas líneas deberán estar relacionadas con el pedido que provocó la compra, de esta manera tendrán un relación con un registro de la tabla *AuxPedido*.

Tabla: País

Atributos: `pai_id`, `pai_nombre`, `pai_zog_id`.

Tabla donde se insertarán todos los países a tener en cuenta para el envío de pedidos y almacenamiento de domicilios de los usuarios. Cada uno de los países pertenece a una zona geográfica (entendidas como continentes).

Tabla: Ciudad

Atributos: `ciu_id`, `ciu_nombre`, `ciu_coordenadas`, `ciu_zonaHoraria`.

Tabla de almacenamiento de ciudades, tendremos en cuenta como atributos tanto su nombre, como coordenadas y zona horaria. Un país podrá tener múltiples ciudades, pero una ciudad únicamente pertenece a un único país.

Tabla: Calle

Atributos: `cal_id`, `cal_nombre`, `cal_codigoPostal`, `cal_ciu_id`.

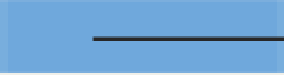


Tabla que almacenará todas las calles que se encuentren dentro de la base de datos, estas deberán de ser de alguna ciudad existente anteriormente. Es así que una calle estará relacionada con una ciudad, pudiendo tener esta múltiples calles, pero una calle únicamente pertenecer a una ciudad.

Tabla: Domicilio

Atributos: `dom_id`, `dom_numero`, `dom_letra`, `dom_bloque`, `dom_cal_id`, `dom_com_id`.

Será nuestra tabla intermedia entre calle y comprador, un domicilio será representado por el número, la letra y bloque en caso de que se viva en uno. Cuando un usuario se registra dentro de la página web a la hora de escoger el domicilio se creará un registro dentro de esta tabla. Dicho domicilio pertenecerá tanto a una calle como a un comprador concreto, permitiendo así que un único comprador tenga múltiples domicilios en diferentes calles.

La multiplicidad de estos registros permitirá que un comprador pueda escoger entre varios domicilios a la hora de hacer su pedido.

Tabla: Comprador

Atributos: `com_id`, `com_dni`, `com_nombre`, `com_apellido1`, `com_apellido2`, `com_usuario`, `com_contraseña`, `com_telefono`.

Se almacenarán registros de usuarios que tengan el rol de comprador, estos tendrán como permisos poder comprar dentro de la página web. De esta manera un comprador será representado con los siguientes atributos: dni, nombre, apellido 1, apellido 2, número de teléfono, usuario y contraseña. Los compradores serán los únicos que podrán llevar a cabo compras en la aplicación web.

Tabla: Pedido

Atributos: `ped_id`, `ped_fecha`, `ped_pagado`, `ped_tarjeta`, `ped_estado`, `ped_com_id`, `ped_dom_id`, `ped_dis_id`, `ped_rep_id`.

Tabla que tendrá como registros el resultado de haber llevado a cabo la acción de comprar un producto. Tendremos como atributos la fecha de realización del pedido, un booleano que indicará si está pagado o no el pedido, el número de la tarjeta bancaria usada para pagar, y el estado del pedido que variará entre los siguientes: Pagado, En proceso de reparto, Entregado al comprador, Comprador no encontrado, Rechazado, Devuelto.

Tabla: Tipo Incidencia

Atributos: tin_id, tin_nombre.

Tabla que contendrá los diferentes tipos de incidencias que pueden resultar de una entrega. Entre los tipos que encontramos están: Pagado, En proceso de reparto, Entregado al comprador, Comprador no encontrado y Rechazado.

Tabla: Incidencia

Atributos: inc_id, inc_titulo, inc_descripcion, inc_tin_id, inc_ped_id, inc_ped_rep_id.

Tabla donde se irán insertando las diferentes incidencias resultantes del reparto del pedido. Será el propio transportista quien notificará las incidencias que puedan ocurrir. En caso de que un pedido tenga 3 veces una incidencia del tipo “Comprador no encontrado” se cambiará el estado del pedido a “Devuelto”. Esta tabla tendrá el *trigger_incidencia*, que será encargado de hacer el conteo de incidencias del tipo “Comprador no encontrado”.

Tabla: Repartidor

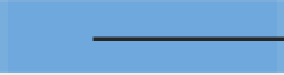
Atributos: rep_id, rep_dni, rep_nombre, rep_apellido1, rep_apellido2, rep_usuario, rep_contraseña, rep_telefono.

Otra de las 4 tablas existentes para identificar usuarios dentro de la página web. El repartidor será el encargado de transportar el pedido a la localización del comprador. Este será asignado a un pedido de manera manual

Tabla: Línea

Atributos: lin_id, lin_cantidad, lin_almacen, lin_diaSinLlegar, lin_ped_id, lin_pro_id.

La tabla de *Línea* hace referencia a los diferentes productos comprados dentro de un mismo pedido, conceptualmente sería las diferentes líneas que puede tener un ticket de compra. Se indicará como atributos de estos registros la cantidad, si ha llegado o no al almacén, y cuántos días lleva sin llegar al almacén de un producto concreto de un pedido.



Esta tabla tendrá un evento llamado *actualizarDiasSinLlegar*, que será el encargado de incrementar el valor de días que han pasado desde la compra y que no ha llegado al almacén.

Tabla: Comentario

Atributos: *cmt_id, cnt_descripcion, cmt_fecha, cmt_pro_id, cmt_com_id*.

Esta tabla almacenará los comentarios de un producto efectuado por un comprador que, en alguno de sus pedidos, haya comprado dicho producto y aún no haya comentado nada sobre él. Se guardará como atributos el cuerpo del mensaje como una descripción y la fecha del mensaje.

Tabla: Vendedor

Atributos: *ven_id, ven_dni, ven_nombre, ven_apellido1, ven_apellido2, ven_usuario, ven_contraseña, ven_telefono, ven_estado*.

Tabla que guardará la información específica de una persona con el rol de vendedor, como pueden ser, a diferencia de la clase Comprador, Controlador y Repartidor, el estado del vendedor que podrá tener los siguientes valores: “FIABLE”, “MALO” y “SOSPECHOSO”. El comportamiento asignado al cambio de estados se encargarán el *trigger_cambiarEstado* y *procedure_cambiarEstado* explicados en el siguiente apartado.

Tabla: Producto

Atributos: *pro_id, pro_nombre, pro_descripcion, pro_medidas, pro_imagen, pro_color, pro_cantidad, pro_precio, pro_oferta, pro_descuento, pro_ven_id*.

En esta tabla se insertarán todos los productos de la página web. Cada producto pertenece a un vendedor, y cada producto tendrá un conjunto de características como: la cantidad disponible, precio, si está de oferta o no, en caso de estar de oferta cuanto es el descuento, medidas, nombre, descripción, imagen y color. Un vendedor podrá tener 1 o más productos.

Tabla: Tipo Producto

Atributos: *tpr_id, tpr_nombre*.

Hace referencia a las categorías de los productos, únicamente almacenaremos los nombres de cada uno de los diferentes tipos que podamos encontrar, ejemplo de datos serían: “Fruta”, “Tecnología”, “Libros”...

Tabla: TipoxProducto

Atributos: txp_id, txp_pro_id, txp_tpr_id.

Tabla intermedia entre Producto y Tipo de producto, con esta tabla se permitirá que un producto tenga múltiples tipos, dando mayor variedad al filtrado por categorías. De esta manera un producto tendrá múltiples tipos, y un tipo podrá aparecer en más de un producto.

Tabla: Controlador

Atributos: con_id, con_nombre, con_apellido1, con_apellido2, con_usuario, con_contraseña.

Tabla del último tipo de usuario que podremos encontrar dentro de la página web, será el encargado de realizar las comprobaciones de llegada de productos al almacén y de notificar a los vendedores en caso de que algún producto de un pedido se haya retrasado 5 días. Entre los atributos que encontramos están: nombre, apellido 1, apellido 2, usuario y contraseña.

Tabla: Aviso

Atributos: avi_id, avi_titulo, avi_descripcion, avi_fecha, avi_con_id, avi_ven_id, avi_lin_id.

Tabla que guardará todos los avisos que hacen los controladores a los vendedores, aquí se almacenarán tanto el título del aviso, descripción del suceso como la fecha del envío del aviso, se hará referencia tanto a la línea que no ha llegado al almacén como del vendedor responsable. Si se han enviado más de 3 avisos y menos de 6 a un vendedor cambiará su estado de “FIABLE” a “SOSPECHOSO”, en el caso de que lleguen 3 más pasará de “SOSPECHOSO” a “MALO”.

6. Funciones, procedures y eventos

Para el desarrollo del proyecto se han creado un conjunto de procedures, functions y triggers para llevar a cabo las funcionalidades solicitadas, es por ello que en esta sección se detalla el funcionamiento de cada uno:

6.1 Procedures

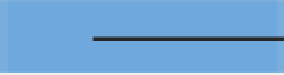
- `procedure_3Avisos`: Procedure que se podrá usar para verificar el cambio de estado de un vendedor al enviar avisos. Dicho procedure insertará 3 avisos sobre una línea de un pedido que ya se encuentra en la base de datos al momento de hacer la importación. De esta manera comprobando sobre el vendedor, que en este caso es el que tenga ID=1, podremos ver el cambio de estado.
- `procedure_cambiarEstado`: Que se ejecutará como consecuencia de un trigger que actúa al insertarse avisos. *trigger_cambiarEstado*. Este procedure se encargará de contar cuantos avisos tiene un vendedor, para que cuando tenga más de 3 avisos y menos de 6 tenga como estado *SOSPECHOSO* y cuando tenga 6 o más, pase a estado de *MALO*.

6.2 Eventos

- `actualizarDiasSinLlegar`: Evento que se ejecutará a las 23:59:59 de cada día y incrementará el contador de días sin llegar al almacén a aquellas líneas que no hayan llegado aún. Como resultado de este evento se podrán llegar a visualizar en 5 días, líneas de productos que aún no hayan llegado dentro de la View de *lineasSinLlegar*. Si quisiéramos comprobar el correcto funcionamiento bastaría con ir dentro de phpmyadmin -> eventos y cambiar el intervalo de ejecución de 1 día a 1 minuto por ejemplo.
- `copiaSeguridad`: Evento que de la misma manera que el anterior se ejecutará a las 23:59:59 todos los días, e insertará dentro de las tablas auxiliares de pedidos y líneas, aquellos registros que se hayan insertado el mismo día (pedidos del día actual junto con sus líneas). Este comportamiento cumplirá que sea una copia incremental de ambas tablas. De la misma manera que con el anterior se podría cambiar el intervalo de ejecución de 1 día, y pasarlo a 1 minuto.

6.3 Triggers

- `trigger_cambiarEstado`: Trigger que se ejecutará cuando se inserte un aviso en la tabla de avisos, lo que dará pie a que luego se haga el conteo y cambio de estado del vendedor con el *procedure_cambiarEstado*.
- `trigger_incidencia`: Trigger que se ejecutará cuando se inserte una incidencia, este contará cuantas incidencias del tipo "Comprador no



encontrado” tiene el pedido del cual se ha hecho la incidencia, en el caso de encontrar 3 se cambiará el estado del pedido a “Devolución”, en cualquier otro caso será el propio del tipo de la incidencia.

7. Vistas

Para que los controladores pudieran observar las líneas de pedidos que no han llegado al almacén en el plazo de 5 días, se ha creado una vista no materializada que tiene como SELECT asociada una consulta que devolverá todo aquel conjunto de líneas que llevan 5 o más días sin llegar y que además tengan el atributo *lin_almacen* a FALSE.

Esta vista, permitirá una fácil visualización para el controlador. Como no estaba incluido dentro de los requisitos la página web que permita el mantenimiento del seguimiento de los pedidos (tarea para equipos de 4 personas), únicamente se ha creado la estructura que permitiría hacer esto mismo eficientemente.

8. Aplicación

Podemos dividir la aplicación en 5 vistas distintas, debido a que gran parte de las funcionalidades asignadas a grupos más grandes no se llevan a cabo:

8.1 Home.php

Vista principal de la aplicación web. Es desde esta página que se podrá ver el catálogo de productos de la base de datos. Podremos llevar a cabo los diferentes tipos de filtrado, tanto por categorías como por nombres de productos.

Los productos presentados tendrán impreso su nombre, descripción, precio y en caso de tener rebajas el precio actualizado. En caso de haber estado logueados como un usuario comprador se nos permitirá comprar dichos productos, en caso contrario no saldrá el botón de comprar. En el caso de que pulsemos dicho producto se guardará dentro del carrito. También si pulsamos encima del nombre del producto, la aplicación nos redirigirá a la pantalla de detalles de productos.

El carrito tendrá el listado de productos que se pretende comprar, además incluye un botón para pasar directamente al formulario de compra. Este carrito se guardará dentro de la barra de navegación y se eliminarán los datos contenidos en él, ya sea manualmente o al cerrar la sesión.

Finalmente desde esta misma página, tendremos 2 botones en la barra de navegación, uno para iniciar sesión y otro para registrarse. En el caso de que hayamos iniciado sesión, estos botones serán reemplazados por el nombre del usuario con el que hemos iniciado sesión y otro botón para cerrar la sesión.

8.2 Details.php

Esta pantalla contendrá todos los datos relacionados con un producto, se mostrarán tanto la imagen en grande, precio, color, medidas y en caso de haber iniciado sesión con un usuario comprador se mostrará adicionalmente las cantidades y botón para añadir al carrito.

Adicionalmente se incorporará una lista de productos similares de otros vendedores.

Finalmente en la parte inferior tendremos un listado de comentarios de otros usuarios comprador. Solo se puede incluir un comentario en dicha lista en el caso de que un comprador haya efectuado correctamente un pedido con ese producto. Además no se permitirá comentar 2 veces sobre el mismo producto

8.3 Checkout.php

Esta página contendrá a modo de resumen el pedido que se va a efectuar. Tendremos por un lado un desplegable que nos permitirá seleccionar a qué dirección se desea enviar el pedido, esta dirección será la misma que la seleccionada a la hora de registrarse, no entra dentro del alcance del proyecto el gestionar la creación de domicilios adicionales, pero en el caso de insertarse manualmente aparecerán en la lista.

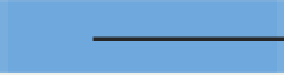
Por otro lado, tenemos un pequeño formulario para hacer el pago del pedido, por lo que deberemos insertar número de la tarjeta, fecha de caducidad y el código de seguridad. De estos datos únicamente se almacenará dentro de la base de datos el número de la tarjeta.

Finalmente en el extremo derecho tendremos una tabla con los diferentes productos almacenados en el carrito, se nos mostrará tanto el nombre del producto, precio, cantidad y el precio total de dicha línea. También se mostrará el precio total del pedido.

Una vez pulsemos sobre el botón de comprar seremos redirigidos a la página principal, el carrito se habrá vaciado y se habrán reducido la cantidad de los productos comprados.

8.4 Register.php

Esta página será donde podremos registrarnos como un usuario dentro de la aplicación. Todos los campos que salgan dentro del formulario son obligatorios de completar salvo el segundo apellido. La selección del rol implicará cambios en los campos del formulario, Por ejemplo un comprador tendrá que rellenar su dirección, pero un controlador no, esto se tratará de manera dinámica con JavaScript.



Por otro lado, en el caso de ser un comprador y seleccionar la dirección, tendremos como valores un listado de países, ciudades y calles (todos insertados anteriormente en la base de datos), por otro lado el número de puerta, letra y número de bloque serán campos rellenables por el usuario, siendo el número de bloque y letra opcionales.

En caso de intentar crear una cuenta de comprador y poner incorrectamente una calle o ciudades que no pertenece a su respectiva ciudad o país, al hacer el registro se nos mostrará un error, debido a que no coinciden la procedencia de las calle, ciudad o país.

Si hiciéramos un registro exitoso de un usuario, nos redirigirá a la pantalla de iniciar sesión.

8.5 Login.php

Esta página contendrá un formulario que permitirá iniciar sesión, necesitaremos indicar tanto el rol, como usuario y contraseña. Cabe aclarar que si hemos registrado un usuario controlador e incorporamos las mismas credenciales para el tipo comprador, nos mostrará un error de que el usuario no existe, eso es debido a que todos los roles se guardan en tablas independientes.

Si conseguimos iniciar sesión correctamente, seremos redirigidos a la pantalla de inicio donde podremos ver los productos y proceder a comprar.

9. Conclusiones

Como conclusiones de la finalización de este proyecto podemos destacar que ha sido una actividad que ha conseguido poner en práctica todo lo aprendido en clase, desde modelo relacional hasta eventos y procedures. Por otro lado, el hecho de implementarlo con PHP, HTML y algo de JavaScript a un nivel bajo, permite ver limitaciones en cuanto a seguridad y tratamiento de datos, por lo que ha sido muy interesante llevar a cabo el proyecto.