[Document title]

[Document subtitle]

[School]

[Course title]

Contents

[1. VISIÓN DE LA APLICACIÓN 1](#_Toc504470910)

[1.1. DESCRIPCIÓN GENERAL 1](#_Toc504470911)

[1.2. ESBOZOS DE LA PÁGINA PRINCIPAL 2](#_Toc504470912)

[1.3. ESQUEMA DE NAVEGACIÓN 3](#_Toc504470913)

[2. ANÁLISIS 4](#_Toc504470914)

[2.1. DIAGRAMA DE CASOS DE USO 5](#_Toc504470915)

[2.2. CASOS DE USO 5](#_Toc504470916)

[3. DISEÑO 14](#_Toc504470917)

[3.1. PERSISTENCIA 15](#_Toc504470918)

[3.2. ENTIDADES 15](#_Toc504470919)

[3.3. INTERACCIÓN 15](#_Toc504470920)

[4. IMPLEMENTACIÓN 17](#_Toc504470921)

[4.1. PÁGINAS PHP 17](#_Toc504470922)

[4.2. ENTRADAS 18](#_Toc504470923)

[4.3. LIBRERÍAS 19](#_Toc504470924)

[4.4. BASE DE DATOS 20](#_Toc504470925)

# VISIÓN DE LA APLICACIÓN

## DESCRIPCIÓN GENERAL

El objetivo de aplicación será vender ordenadores (de mesa y portátiles) y móviles a medida, es decir, el usuario podrá elegir todos los componentes del dispositivo, desde el tipo de pantalla que tendrá hasta los programas que vendrán preinstalados, pero no se podrán comprar directamente artículos a medida previamente configurados ni comprar componentes por separado.

Una vez dentro de la página, el usuario podrá elegir qué tipo de producto desea comprar. Cuando lo haya elegido, le aparecerán distintas opciones para personalizar todos sus componentes (placa base, procesador o si desea un Jack de audio, entre otros). Los usuarios podrán utilizar la aplicación, es decir, navegar por los artículos y configurar productos, sin estar registrados, pero deben estarlo para realizar compras.

La aplicación contará con una base de datos en la que se guardará la información sobre los componentes, pedidos y usuarios, así como de un tipo de usuario administrador para controlar esta base de datos. El usuario administrador será el encargado de crear, borrar y modificar los productos que se venden en la aplicación. El administrador no podrá acceder a los datos de los clientes ni los pedidos. Los clientes podrán gestionar los datos de su cuenta, los métodos de pago, y podrán consultar su historial de pedidos.

## ESBOZOS DE LA PÁGINA PRINCIPAL



Figura 1: Página principal



Figura 1: Construcción de un ordenador

## ESQUEMA DE NAVEGACIÓN



# ANÁLISIS

## DIAGRAMA DE CASOS DE USO



## CASOS DE USO

**CU01: LOGIN**

**Descripción:** El usuario se loguea para acceder a funcionalidades avanzadas

**Actores:** Usuario

**Precondición:** Ninguna

**Postcondiciones:** El sistema admite al usuario

**Punto de extensión:** CU02: Registrarse

**Flujo normal:**

1. El usuario inicia un nuevo proceso de login.
2. El sistema solicita el usuario y la contraseña.
3. El usuario introduce el usuario y la contraseña.
4. El sistema devuelve al usuario a la operación que estaba realizando.

**Flujos alternativos:**

3a. Datos del formulario incorrectos

1. El sistema informa del error y vuelve a mostrar los formularios

Punto de extensión: Línea 2: El usuario inicia el CU02: Registrarse

**CU02: REGISTRARSE**

**Descripción:** El usuario se registra para darse de alta en el servicio

**Actores:** Usuario

**Precondición:** Ninguna

**Postcondiciones:** El sistema da de alta al usuario

**Flujo normal:**

1. El usuario inicia un nuevo proceso de registro.
2. El sistema muestra el formulario con los datos requeridos.
3. El usuario rellena el formulario.
4. El sistema valida el formulario y devuelve al usuario a la operación que estaba realizando.

**Flujos alternativos:**

3a. Datos del formulario incorrectos

1. El sistema informa del error y vuelve a mostrar los formularios

**CU03: BUSCAR PRODUCTO**

**Descripción:** El administrador busca un producto en el sistema

**Actores:** Usuario

**Precondición:** Ninguna

**Postcondiciones:** El sistema devuelve el producto encontrado

**Flujo normal:**

1. El usuario inicia un nuevo proceso de búsqueda de producto.
2. El sistema solicita el parámetro de búsqueda.
3. El usuario introduce el parámetro de búsqueda.
4. El sistema devuelve el resultado de la búsqueda.

**CU04: VER CUENTA**

**Descripción:** El cliente visualiza los detalles de su cuenta

**Actores:** Cliente

**Precondición:** El usuario debe haber iniciado sesión mediante el CU01: Login

**Punto de extensión:** CU05: Ver Pedidos, CU06: Actualizar Cuenta, CU15: Borrar Cuenta, CU07: Añadir Método de Pago, CU08: Borrar Método de Pago, CU09: Actualizar Método de Pago

**Flujo normal:**

1. El Cliente desea ver los detalles de su cuenta.
2. El sistema muestra los detalles de la cuenta del cliente.

Punto de extensión 1: Línea 2: El cliente inicia el CU05: Ver Pedidos

Punto de extensión 2: Línea 2: El cliente inicia el CU06: Actualizar Cuenta

Punto de extensión 3: Línea 2: El cliente inicia el CU15: Borrar Cuenta

Punto de extensión 4: Línea 2: El cliente inicia el CU07: Añadir Método de Pago

Punto de extensión 5: Línea 2: El cliente inicia el CU08: Borrar Método de Pago

Punto de extensión 6: Línea 2: El cliente inicia el CU09: Actualizar Método de Pago

Punto de extensión 7: Línea 2: El cliente inicia el CU16: Añadir Dirección

Punto de extensión 8: Línea 2: El cliente inicia el CU17: Borrar Dirección

Punto de extensión 9: Línea 2: El cliente inicia el CU18: Actualizar Dirección

**CU05: VER PEDIDOS**

**Descripción:** El cliente visualiza los pedidos de su cuenta

**Actores:** Cliente

**Precondición:** El usuario debe haber iniciado sesión mediante el CU01: Login

**Flujo normal:**

1. El Cliente desea ver los pedidos de su cuenta.
2. El sistema muestra los pedidos de la cuenta del cliente.

**CU06: ACTUALIZAR CUENTA**

**Descripción:** El cliente actualiza los datos de su cuenta.

**Actores:** Cliente

**Precondición:** El usuario debe haber iniciado sesión mediante el CU01: Login, y haber accedido a la información de su cuenta mediante el CU04: Ver cuenta,

**Postcondición:** Se actualizan los datos de la cuenta del cliente.

**Flujo normal:**

1. El Cliente desea actualizar los datos de su cuenta.
2. El Sistema pide los nuevos datos de la cuenta.
3. El Cliente rellena el formulario con los nuevos datos.
4. El Sistema registra los cambios realizados en la cuenta del cliente.

**Flujos alternativos:**

Línea 3. Datos incorrectos.

1. El sistema informa del error y vuelve a mostrar el formulario.

**CU07: AÑADIR MÉTODO DE PAGO**

**Descripción:** El cliente agrega un método de pago a su cuenta.

**Actores:** Cliente

**Precondición:** El usuario debe haber iniciado sesión mediante el CU01: Login, y haber accedido a la información de su cuenta mediante el CU04: Ver cuenta.

**Postcondición:** Se agrega un nuevo método de pago en la cuenta del cliente.

**Flujo normal:**

1. El Cliente desea agregar un nuevo método de pago a su cuenta.
2. El sistema muestra los campos a rellenar de tipo, fecha de caducidad, número y código de seguridad del método de pago.
3. El cliente rellena los campos con los datos del nuevo método de pago.
4. El Sistema registra el método de pago en la cuenta del cliente.

**Flujos Alternativos:**

Línea 3. Cuenta caducada.

1. El sistema informa del error y vuelve a la línea 2.

Línea 3. Datos incorrectos.

1. El sistema informa del error y vuelve a la línea 2.

**CU08: BORRAR MÉTODO DE PAGO**

**Descripción:** El cliente elimina un método de pago a su cuenta.

**Actores:** Cliente

**Precondición:** El usuario debe haber iniciado sesión mediante el CU01: Login, y haber accedido a la información de su cuenta mediante el CU04: Ver cuenta.

**Postcondición:** Se elimina un método de pago en la cuenta del cliente.

**Flujo normal:**

1. El Cliente desea borrar un método de pago a su cuenta.
2. El sistema muestra los métodos de pago que hay en la cuenta del cliente.
3. El cliente selecciona uno de los métodos de pago.
4. El Sistema borra el método de pago en la cuenta del cliente.

**CU09: ACTUALIZAR MÉTODO DE PAGO**

**Descripción:** El cliente actualiza un método de pago a su cuenta.

**Actores:** Cliente

**Dependencias:** CU04: Ver cuenta

**Precondición:** El usuario debe haber iniciado sesión mediante el CU01: Login, y haber accedido a la información de su cuenta mediante el CU04: Ver cuenta.

**Postcondición:** Se actualiza un método de pago en la cuenta del cliente.

**Flujo normal:**

1. El Cliente desea actualizar un método de pago a su cuenta.
2. El Sistema muestra los métodos de pago que hay en la cuenta del cliente.
3. El Cliente selecciona uno de los métodos de pago.
4. El Sistema pide los nuevos datos de la cuenta.
5. El Cliente rellena los nuevos datos.
6. El Sistema registra los cambios realizados en el método de pago del cliente.

**Flujos alternativos:**

Línea 5. Cuenta caducada.

1. El sistema informa del error y vuelve a la línea 4.

Línea 5. Datos incorrectos.

1. El sistema informa del error y vuelve a la línea 4.

**CU10: CONSTRUIR PRODUCTO**

**Descripción:** El cliente realiza un pedido de un producto construido a piezas.

**Actores:** Cliente

**Precondición:** Ninguna.

**Postcondición:** Se registra un nuevo pedido.

**Flujo normal:**

1. El Cliente desea comprar un producto construido a piezas.
2. El Sistema pide el tipo de producto que se quiere construir.
3. El Cliente selecciona el producto.
4. El Sistema pide que se seleccione un componente del producto.
5. El Cliente selecciona un componente del producto.

Se repiten las líneas 4 – 5 hasta que el cliente seleccione todos los componentes del producto.

1. El Sistema añade el producto a la cesta

**CU11: REALIZAR PEDIDO**

**Descripción:** El cliente procesa el pedido de su cesta.

**Actores:** Cliente

**Precondición:** El cliente debe haber añadido al menos 1 artículo a la cesta mediante el CU10: Construir Producto.

**Postcondición:** Se procesa un nuevo pedido.

**Flujo normal:**

1. El Cliente inicia un proceso de realización de pedido.
2. El Sistema muestra los artículos del pedido y solicita una dirección de envío y un método de pago.
3. El Cliente introduce la dirección de envío y el método de pago y confirma el pedido.
4. El Sistema registra los cambios.

**CU12: AÑADIR PRODUCTO**

**Descripción:** El administrador añade un producto al sistema

**Actores:** Administrador

**Dependencias:**  Ninguna

**Precondición:** El administrador debe haberse logueado mediante el CU01: Login

**Postcondiciones:** La información del producto se actualiza

**Flujo normal:**

1. El administrador inicia un nuevo proceso de adición de producto.
2. El sistema muestra el formulario con los datos del nuevo producto.
3. El administrador rellena el formulario con los datos del nuevo producto.
4. El sistema registra los cambios y muestra los nuevos datos.

**Flujos alternativos:**

3a. Datos del formulario incorrectos

1. El sistema informa del error y vuelve a mostrar los formularios

**CU13: ACTUALIZAR PRODUCTO**

**Descripción:** El administrador actualiza la información sobre uno de los productos

**Actores:** Administrador

**Dependencias:**  CU03: Buscar producto

**Precondición:** El administrador debe haberse logueado mediante el CU01: Login

**Postcondiciones:** La información del producto se actualiza

**Flujo normal:**

1. El administrador inicia un nuevo proceso de adición de producto.
2. El sistema solicita los datos del nuevo producto.
3. El administrador rellena el formulario con los datos del nuevo producto.
4. El sistema registra los cambios.

**Flujos alternativos:**

3a. Datos del formulario incorrectos

1. El sistema informa del error y vuelve a mostrar los formularios

**CU14: BORRAR PRODUCTO**

**Descripción:** El administrador borra un producto de la base de datos

**Actores:** Administrador

**Dependencias:**  CU03: Buscar producto

**Precondición:** El administrador debe haberse logueado mediante el CU01: Login

**Postcondiciones:** El producto desaparece de la base de datos

**Flujo normal:**

1. El administrador inicia un nuevo proceso de borrado de producto.
2. Se inicia el CU03: Buscar producto y devuelve el producto encontrado.
3. El administrador borra el producto.
4. El sistema registra los cambios.

**Flujos alternativos:**

2a. No se encuentra el producto

1. El sistema informa del error y vuelve a iniciar el CU03

**CU15: BORRAR CUENTA**

**Descripción:** El cliente elimina su cuenta y toda la información.

**Actores:** Cliente

**Precondición:** El usuario debe haber iniciado sesión mediante el CU01: Login, y haber accedido a la información de su cuenta mediante el CU04: Ver cuenta.

**Postcondición:** Se elimina un usuario del sistema.

**Flujo normal:**

1. El Cliente desea borrar su cuenta.
2. El sistema solicita la contraseña del cliente.
3. El cliente proporciona su contraseña.
4. El Sistema borra al cliente.

**Flujos alternativos:**

3a. Contraseña incorrecta

1. El sistema informa del error y cancela la operación.

**CU16: AÑADIR DIRECCIÓN**

**Descripción:** El cliente agrega una dirección a su cuenta.

**Actores:** Cliente

**Precondición:** El usuario debe haber iniciado sesión mediante el CU01: Login, y haber accedido a la información de su cuenta mediante el CU04: Ver cuenta.

**Postcondición:** Se agrega una nueva dirección en la cuenta del cliente.

**Flujo normal:**

1. El Cliente desea agregar una nueva dirección a su cuenta.
2. El sistema muestra los campos a rellenar.
3. El cliente rellena los campos con los datos de la nueva dirección.
4. El Sistema registra la dirección en la cuenta del cliente.

**Flujos Alternativos:**

Línea 3. Datos incorrectos.

1. El sistema informa del error y vuelve a la línea 2.

**CU17: BORRAR DIRECCIÓN**

**Descripción:** El cliente elimina una dirección de su cuenta.

**Actores:** Cliente

**Precondición:** El usuario debe haber iniciado sesión mediante el CU01: Login, y haber accedido a la información de su cuenta mediante el CU04: Ver cuenta.

**Postcondición:** Se elimina una dirección en la cuenta del cliente.

**Flujo normal:**

1. El Cliente desea borrar una dirección de su cuenta.
2. El sistema muestra las direcciones que hay en la cuenta del cliente.
3. El cliente selecciona una de las direcciones.
4. El Sistema borra una dirección de la cuenta del cliente.

**CU18: ACTUALIZAR DIRECCIÓN**

**Descripción:** El cliente actualiza una dirección de su cuenta.

**Actores:** Cliente

**Dependencias:** CU04: Ver cuenta

**Precondición:** El usuario debe haber iniciado sesión mediante el CU01: Login, y haber accedido a la información de su cuenta mediante el CU04: Ver cuenta.

**Postcondición:** Se actualiza una dirección en la cuenta del cliente.

**Flujo normal:**

1. El Cliente desea actualizar una dirección de su cuenta.
2. El Sistema muestra las direcciones que hay en la cuenta del cliente.
3. El Cliente selecciona una de las direcciones.
4. El Sistema pide los nuevos datos de la cuenta.
5. El Cliente rellena los nuevos datos.
6. El Sistema registra los cambios realizados en la dirección del cliente.

**Flujos alternativos:**

Línea 5. Datos incorrectos.

1. El sistema informa del error y vuelve a la línea 4.

QAEQWE

# DISEÑO

## PERSISTENCIA

La aplicación cuenta con una base de datos cuya función es encargarse de la persistencia de la información relevante. Las entidades de esta base de datos se organizan según el siguiente diagrama entidad-relación:



## ENTIDADES

1. Usuario: refleja a una persona que hace uso de la aplicación: Hay dos tipos de usuario, clientes y administradores. Los administradores tienen acceso a funcionalidades avanzadas.
2. Método de pago: Cada usuario tiene una serie de métodos de pago con los que realizar los pedidos. Éstos se almacenan en la base de datos para que sea más conveniente a la hora de realizar los pagos.
3. Dirección: Cada usuario tiene también una serie de direcciones a las que se pueden realizar los pedidos. De igual modo, se almacenan en la base de datos para facilitar la conveniencia de los usuarios.
4. Pedido: Los clientes pueden realizar pedidos, que se asocian con su entidad. Cada pedido debe guardar información sobre a qué dirección se envía y con qué método de pago se va a completar, de entre los que un cliente tenga disponible.
5. Componente: Los componentes se emplean para construir ordenadores o teléfonos móviles. Sólo los usuarios administradores pueden crear, borrar o modificar componentes.
6. Producto montable: Refleja estos ordenadores y móviles que los usuarios personalizan con los componentes. Un pedido se compone de uno o más de estos productos montables.

## INTERACCIÓN

Para la interacción entre el usuario y la base de datos, y así como el procesamiento de la información, se ha empleado un intérprete PHP alojado en un servidor web Apache. Este intérprete permite recoger los datos de entrada por parte de los usuarios mediante formularios, procesar la información, y hacer las operaciones pertinentes con la base de datos. Después, los resultados se muestran a los usuarios en forma de documento HTML. Estas funcionalidades PHP se dividen en diferentes ficheros, donde cada uno realiza una de ellas.

MÓDULOS:

1. Account: Se encarga de gestionar todo lo relacionado con los usuarios, excepto la creación. Desde aquí, se pueden visualizar y modificar los atributos de los usuarios, así como visualizar sus pedidos (con sus productos montables), direcciones y métodos de pago, y acceder a la página que se encarga de modificar estos dos últimos. También permite a un usuario borrar su cuenta.
2. Addresses: Se encarga de gestionar una dirección de un usuario. Mediante este módulo, podemos crear, borrar o modificar la dirección que se haya seleccionado en Account. No se pueden modificar o eliminar varias direcciones a la vez.
3. PaymentMethods: Se encarga de gestionar un método de pago de un usuario. Mediante este módulo, podemos crear, borrar o modificar el método de pago que se haya seleccionado en Account. No se pueden modificar o eliminar varios métodos de pago a la vez.
4. Register: Se encarga exclusivamente de la creación de usuarios.
5. ManageProduct: Permite a los usuarios administradores crear, borrar y modificar productos. A este módulo se accede a través de Account con un usuario administrador.
6. NewProduct: Se encarga de crear productos montables.
7. Order: Se encarga de crear pedidos con los productos montables completados.

# IMPLEMENTACIÓN

## PÁGINAS PHP



Además del esquema que aquí se muestra, existen tres módulos más:

1. functions.php: Fichero PHP que recoge funciones de utilidad que se emplean en todas las demás páginas, pero al cual no se puede acceder.
2. header.php: Módulo que se incluye en la cabecera de todas las páginas para generar un header común.
3. footer.php: Módulo que se incluye al final de todas las páginas para generar un footer común.

Es necesario mencionar que, en cualquier momento, el usuario puede realizar las siguientes operaciones, independientemente de donde se encuentre:

1. Volver al index.php haciendo clic en el icono del header.
2. Loguearse o registrarse haciendo clic en el botón apropiado del header.
3. Ver su cuenta, en caso de que esté logueado, usando el botón apropiado del header.
4. Ver su carrito, en caso de que contenga algún artículo, usando el botón apropiado del header.

Siempre que un usuario sea direccionado a login.php, tiene la opción de registrarse si no lo está. Una vez registrado, el sistema le devolverá a la operación que estuviera realizando:



## ENTRADAS

Una vez se haya logueado o registrado un usuario a través de login.php o register.php, el servidor web guardará su correo, user\_id, y si es un administrador, mediante una variable de sesión, que seguirá activa hasta que el usuario cierre la sesión o el navegador. Así, el resto de las páginas de la aplicación (en concreto, account.php, addresses.php, paymentMethods.php y orders.php) pueden hacer uso de este valor en cualquier momento para realizar sus consultas.

1. account.php: Esta página incorpora muchos intercambios de información, en concreto:
   1. A addresses.php se pasa, por medio de formulario post, si el usuario quiere modificar o borrar una dirección (junto con los datos de la misma, guardado en una variable de sesión), y, en otro formulario, si el usuario ha seleccionado crear una.
   2. A paymentMethods.php se pasa, por medio de formulario post, si el usuario quiere modificar o borrar un nuevo método de pago (junto con los datos del mismo, guardado en una variable de sesión), y, en otro formulario, si el usuario ha seleccionado crear uno.
   3. A sí misma, se pasa, por medio de formulario post, los nuevos datos del cliente cuando este los modifica. También con un formulario post, si el usuario elige borrar su cuenta.
2. addresses.php: Se pasa, a sí misma, un formulario post con los datos introducidos por el usuario para su validación.
3. paymentMethods.php: Se pasa, a sí misma, un formulario post con los datos introducidos por el usuario para su validación.
4. manageProduct.php: Tiene dos formularios post con los que el administrador puede interactuar con esta misma página:
   1. Un formulario post para crear un nuevo producto.
   2. Un formulario post para modificar un producto ya existente. La página genera uno por producto.
5. index.php: Contiene dos formularios post que indican a newProduct.php si el usuario desea construir un ordenador o un móvil.
6. newProduct.php: Contiene dos formularios post, uno para ordenadores y uno para móviles, aunque nunca se muestran los dos a la vez. Los datos de entrada de este formulario se guardan en una variable de sesión con la que se implementa la funcionalidad del carrito. Esta variable de sesión perdura hasta que el usuario cierre la sesión o el navegador.
7. orders.php: Emplea la variable de sesión creada en la página anterior para mostrar un resumen del pedido y pedir confirmación al usuario. Esta confirmación se envía por formulario get a sí misma para procesar los datos.
8. register.php: Emplea un formulario post, enviado a sí misma, para validar y procesar los datos introducidos por el usuario.
9. header.php: Emplea las variables de sesión de usuario y carrito para mostrar los botones apropiados con los que acceder a dichas funcionalidades. Aquí es donde se controla que un usuario no logueado no pueda acceder a su cuenta, o que no se pueda manipular un carrito vacío.

## LIBRERÍAS

El fichero functions.php contiene una librería con funciones de utilidad empleadas en diversas páginas de la aplicación:

|  |  |  |  |
| --- | --- | --- | --- |
| Función | Propósito | Argumentos | Valor devuelto |
| doesCountryExist | Comprueba que el país introducido por el usuario existe realmente | Código ISO ALPHA-3 del país | Verdadero o falso |
| getCountryName | Nombre completo de un país a partir de un código | Código ISO ALPHA-3 del país | Nombre del país. |
| printErrorMessage | Imprime un mensaje de error de servidor | Vector de mensajes de error en texto plano | Vector de mensajes de error formateados |
| createConnection | Crea una conexión a la base de datos | Ninguno | Enlace a la base de datos |
| addressDetails | Devuelve las direcciones asociadas al usuario logueado | Ninguno | Vector de direcciones |
| getAddressForOrder | Devuelve la dirección con un determinado id | Id de la dirección | Dirección con ese id |
| paymentMethodDetails | Devuelve los métodos de pago asociadas al usuario logueado | Ninguno | Vector de métodos de pago |
| validPaymentMethods | Comprueba si un método de pago es válido | Método de pago | Método de pago, si es válido. |
| checkSession | Comprueba si el usuario está logueado, y lo dirige al login si no lo está | Ninguno | Ninguno |
| printWelcome | Imprime la bienvenida y los botones apropiados del header según las variables de sesión | Ninguno | Ninguno |
| getAllComponents | Devuelve todos los componentes de la base de datos | Ninguno | Vector de componentes |

## BASE DE DATOS

La base de datos se ha implementado con el siguiente modelo relacional:



Cada entidad del diseño entidad-relación tiene su tabla, y cada atributo, una columna. Se han relacionado de la siguiente manera:

1. Relaciones muchos-a-muchos: se relacionan creando una tabla intermedia
   1. users y address se relacionan con la tabla user\_address, que guarda un id de cada una de las dos entidades.
   2. custom\_products y components se relacionan con la tabla custom\_products\_components del mismo modo.
2. Relaciones uno-a-muchos:
   1. users y payment\_methods: Un usuario tiene varios métodos de pago, por lo que cada método de pago tiene una clave foránea (FK) del usuario al que pertenece.
   2. orders y custom\_products: Un pedido tiene uno o más productos montables, por lo que estos últimos tienen una FK del pedido al que pertenecen.
   3. payment\_methods, addresses y orders: Un método de pago y una dirección son necesarios para un pedido, por lo que este guarda una FK de cada uno de los anteriores.

Esta implementación se ha desplegado sobre un servidor MySQL 5.3.