



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Departamento de Informática de Sistemas y Computadores

Adaptación de un Benchmark para e-commerce  
bajo plataformas actuales de ventas como  
Prestashop haciendo uso del generador de carga  
GUERNICA.

Trabajo Fin de Master

**Master en Ingeniería de Computadores y Redes**

**Autor:** Luis García Gómez

**Tutores:** Ana Pont Sanjuán

José A. Gil Salinas

8 de septiembre de 2017



# Agradecimientos

---

Agradecer a mis tutores Ana y José Antonio la oportunidad que me han brindado de trabajar con ellos, al igual que agradecerles toda la ayuda y soporte que me han proporcionado en este año.

También, agradecer a mi familia y amigos por estar siempre ahí, dándome fuerzas, apoyándome, aguantándome en momentos difíciles y a veces, con mi humor de Gremlin malo.

Por último, este trabajo está dedicado a mis sobrinas Blanca y Lucía, a las que quiero mucho, además de a mis padres. Siempre os llevare conmigo, siempre fuerte como me enseñasteis y estéis donde estéis siempre amor eterno por vosotros. Se os echa de menos.



# Resumen

---

La tendencia actual es el hacer compras vía online, desde un simple reloj hasta unos muebles para el salón, entramos en Ebay, vemos ofertas de Amazon, compramos algún libro o tal vez algo de ropa, y así un sinnúmero de consultas diarias, ventas y miles de bits trasladados de una parte a otra. Se ha evolucionado y la gente ya no compra solamente yendo a las tiendas, si no las tiendas vienen a casa o están en nuestros Smartphones y Tablets. Las tiendas electrónicas esperan a que los usuarios compren con la mayor facilidad posible, intentando dar el mejor servicio que puedan ofrecer para convertir un mayor número de visitas en ventas y así, generar mayores beneficios. De esta forma, el auge de los e-commerce en la actualidad es algo inminente y necesita de un estudio exhaustivo ya que mueve millones de euros al año, evolucionando muy rápido, debido a los cambios tecnológicos y a que la demanda, a veces, es mayor que la oferta y de esta forma, con buen reclamo, y una navegación óptima del usuario, puede traducirse en muchos euros para el vendedor, con el consiguiente beneficio económico para el proveedor de servicios también.

Dado que hoy en día no existen muchos benchmarks para web y más en concreto para e-commerce, en este proyecto se pretende adaptar un estándar de benchmarking para e-commerce creado por el Transfer Protocol Consortium (TPC), el cual se encuentra obsoleto en la actualidad, a una plataforma de ventas actual como es Prestashop, haciendo uso del generador de carga GUERNICA desarrollado por el grupo de investigación.

Lo que se desea conseguir es, respetando el estándar TPC-W y mediante GUERNICA, comprobar que las navegaciones que se producen sobre una página web para comercio electrónico bajo Prestashop haciendo uso de GUERNICA, se ajustan a los estándares de la especificación de TPC-W, obteniendo una adaptación de dicho estándar a una plataforma actual para e-commerce hoy en día.

La adaptación final es buena siendo necesarios unos retoques de refinamiento. Todo funciona de forma correcta y se han realizado experimentos obteniendo resultados exitosos. Aun así, faltarían algunos puntos que ajustar para obtener mejores resultados y poder trabajar con esta adaptación como un benchmark válido y factible para poner a prueba un e-commerce real o una infraestructura cloud, que ofrezca esta plataforma de explotación de servicios.

**Palabras clave:** Benchmarking, E-Commerce, Prestashop, TPC-W, GUERNICA, TGI, Navegación web.

# Abstract

---

The current trend is to shop online, from a simple clock to a living room furniture, we enter Ebay, we see Amazon offers, we buy some books or maybe some clothes, and so endless daily queries, sales and thousands of bits transferred from one part to another. It has evolved and people no longer buy just by going to stores, otherwise stores come home or are on our Smartphones and Tablets. Electronic stores are waiting for users to buy as easily as possible, trying to give the best service they can offer to convert a larger number of visits to sales and thus generate greater profits. In this way, the e-commerce boom is currently imminent and needs a thorough study since it moves millions of euros a year, evolving very fast, due to technological changes since the demand, sometimes, is greater than the offer and thus, with good claim, and an optimal navigation of the user, can translate into many euros for the seller, with the consequent economic benefit to the service provider also.

Given that today there are not many benchmarks for the web and more specifically for e-commerce, this project aims to adapt an e-commerce benchmarking standard created by the Transfer Protocol Consortium (TPC), which is obsolete at present, to a current sales platform such as Prestashop, making use of the GUERNICA load generator developed by the research group.

What we want to achieve is, according to the TPC-W standard and through GUERNICA, to verify that the navigations that occur in a webpage for electronic commerce under Prestashop making use of GUERNICA, comply with the standards of the TPC- W, obtaining an adaptation of this standard to a current platform for electronic commerce until the current date.

The final adaptation is good and some refinements are necessary. Everything works well and experiments have been carried out to get successful results. Even so, there will be some missing points to adjust to better results and be able to work with this adaptation as a valid and feasible benchmark to test a real e-commerce or cloud infrastructure offered by this service exploitation platform.

**Keywords:** Benchmarking, E-Commerce, Prestashop, TPC-W, GUERNICA, TGI, Web Navigation.

# Tabla de contenidos

---

1	Introducción.....	1
1.1.	Motivación.....	1
1.2.	Objetivos.....	1
1.3.	Estructura del documento.....	3
2	Estado del arte .....	4
2.1.	Panorama actual de la web y los e-commerce .....	4
2.2.	El uso de CMS's.....	5
3	TPC-W.....	10
3.1.	Descripción y trabajos previos.....	10
3.2.	Conceptos importantes .....	11
3.3.	Errores encontrados.....	17
3.4.	Soluciones adoptadas.....	20
3.5.	Validación de los cambios efectuados.....	25
4	Prestashop .....	31
4.1.	Descripción.....	31
4.2.	Instalación .....	32
4.3.	Popularización de la base de datos.....	34
4.4.	Cambios necesarios .....	39
4.5.	Resultado final .....	40
5	GUERNICA.....	41
5.1.	Descripción.....	41
5.2.	Navegaciones.....	41
5.3.	Validación de las navegaciones.....	48
6	Experimentos .....	53
6.1.	Descripción de los experimentos.....	53
6.2.	Datos obtenidos .....	53
6.3.	Análisis de resultados .....	54
7	Conclusiones y trabajos futuros .....	65
7.1.	Conclusiones .....	65
7.2.	Trabajos futuros.....	66
	Bibliografía.....	67

# Tabla de figuras

---

FIGURA 1: CMS'S POR ÁMBITO DE EMPLEO .....	6
FIGURA 2: SOLUCIONES CMS A NIVEL GLOBAL.....	7
FIGURA 3: PRESENCIA E-COMMERCE A NIVEL MUNDIAL.....	8
FIGURA 4: PRESENCIA E-COMMERCE EN ESPAÑA.....	9
FIGURA 5: BASE DE DATOS TPC-W .....	11
FIGURA 6: FACTOR DE ESCALA DE LA BASE DE DATOS TPC-W .....	12
FIGURA 7: EJEMPLO DE UN CUSTOMER BEHAVIOR MODEL GRAPH (CBMG) .....	13
FIGURA 8: GRAFO GENERAL DE TPC-W .....	14
FIGURA 9: PORCENTAJES DE NAVEGACIONES WEB .....	15
FIGURA 10: ASPECTO DE LOS ERRORES DE NOMBRE, APELLIDOS Y TELÉFONO DE TPC-W .....	18
FIGURA 11: ASPECTO DE LOS ERRORES DE EMAIL Y DESCUENTO DE TPC-W.....	18
FIGURA 12: ASPECTO DE UNA BÚSQUEDA EN TPC-W .....	19
FIGURA 13: ERROR DE TPC-W AL REALIZAR UNA BÚSQUEDA.....	19
FIGURA 14: PETICIONES WEB DE LOS USUARIOS PARA EL BROWSING MIX (BROWSER).....	25
FIGURA 15: PETICIONES WEB DE LOS USUARIOS PARA EL BROWSING MIX (ORDER) .....	26
FIGURA 16: PETICIONES WEB DE LOS USUARIOS PARA EL SHOPPING MIX (BROWSER).....	26
FIGURA 17: PETICIONES WEB DE LOS USUARIOS PARA EL SHOPPING MIX (ORDER) .....	27
FIGURA 18: PETICIONES WEB DE LOS USUARIOS PARA EL ORDERING MIX (BROWSER) .....	27
FIGURA 19: PETICIONES WEB DE LOS USUARIOS PARA EL ORDERING MIX (ORDER) .....	28
FIGURA 20: WIPS EN TPC-W SIN CAMBIOS .....	28
FIGURA 21: WIPS EN TPC-W CON CAMBIOS.....	29
FIGURA 22: TIEMPO DE PENSAR SIN CAMBIOS EN TPC-W .....	29
FIGURA 23: TIEMPO DE PENSAR CON CAMBIOS EN TPC-W.....	30
FIGURA 24: PRESENCIA ACTUAL DE LAS PRINCIPALES PLATAFORMAS DE SERVIDORES .....	31
FIGURA 25: PANEL DE CONTROL DE PRESTASHOP .....	33
FIGURA 26: FRONTEND DE PRESTASHOP .....	33
FIGURA 27: BASE DE DATOS DE PRESTASHOP.....	34
FIGURA 28: CLIENTES DESPUÉS DE IMPORTAR EL ARCHIVO CSV .....	35
FIGURA 29: DIRECCIONES DESPUÉS DE IMPORTAR SU ARCHIVO CSV.....	36
FIGURA 30: AUTORES DESPUÉS DE IMPORTAR SU ARCHIVO CSV .....	37
FIGURA 31: PRODUCTO DESPUÉS DE IMPORTAR SU ARCHIVO CSV .....	38
FIGURA 32: ASPECTO DE UN PRODUCTO EN LA PÁGINA WEB DE PRESTASHOP .....	39
FIGURA 33: VALIDACIÓN DE LOS 10 USUARIOS CONECTADOS PRESTASHOP .....	49
FIGURA 34: VALIDACIÓN DE LOS 20 USUARIOS CONECTADOS EN PRESTASHOP.....	50

FIGURA 35: VALIDACIÓN DE REGISTRO DE USUARIOS EN PRESTASHOP.....	50
FIGURA 36: VALIDACIÓN PEDIDOS REALIZADOS PRESTASHOP.....	51
FIGURA 37: VALIDACIÓN HACIENDO USO DE LA CONSOLA DE ECLIPSE.....	52
FIGURA 38: SHOPPING MIX 10 USUARIOS (BROWSE).....	55
FIGURA 39: SHOPPING MIX 10 USUARIOS (ORDER).....	56
FIGURA 40: SHOPPING MIX 20 USUARIOS (BROWSER).....	56
FIGURA 41: SHOPPING MIX 20 USUARIOS (ORDER).....	57
FIGURA 42: SHOPPING MIX 30 USUARIOS (BROWSER).....	57
FIGURA 43: SHOPPING MIX 30 USUARIOS (ORDER).....	58
FIGURA 44: BROWSING MIX 10 USUARIOS (BROWSER).....	58
FIGURA 45: BROWSING MIX 10 USUARIOS (ORDER).....	59
FIGURA 46: BROWSING MIX 20 USUARIOS (BROWSER).....	59
FIGURA 47: BROWSING MIX 20 USUARIOS (ORDER).....	60
FIGURA 48: BROWSING MIX 30 USUARIOS (BROWSER).....	60
FIGURA 49: BROWSING MIX 30 USUARIOS (ORDER).....	61
FIGURA 50: ORDERING MIX 10 USUARIOS (BROWSER).....	61
FIGURA 51: ORDERING MIX 10 USUARIOS (ORDER).....	62
FIGURA 52: ORDERING MIX 20 USUARIOS (BROWSE).....	62
FIGURA 53: ORDERING MIX 20 USUARIOS (ORDER).....	63
FIGURA 54: ORDERING MIX 30 USUARIOS (BROWSER).....	63
FIGURA 55: ORDERING MIX 30 USUARIOS (ORDER).....	64

# Índice de tablas

---

TABLA 1: CARACTERÍSTICAS DE LAS PÁGINAS WEB DE TPC-W .....	16
TABLA 2: UMBRALES PARA EL INTERVALO DE COMPRAS .....	42
TABLA 3: UMBRALES PARA EL INTERVALO DE COMPRAS .....	43
TABLA 4: UMBRALES PARA EL INTERVALO DE COMPRAS .....	43
TABLA 5: RESULTADOS OBTENIDOS PARA SHOPPING MIX CON 10 USUARIOS EN PRESTASHOP .....	54

# 1 Introducción

---

## 1.1. Motivación

En la actualidad el uso de aplicaciones web para realizar compras es cada vez mayor, siendo una importante fuente de ingresos tanto para el sector de los vendedores, como para el sector que ofrece este tipo de alojamientos en infraestructuras cloud o servidores dedicados. Además, el incremento de clientes en los últimos años es masivo, siendo uno de los principales canales de venta hoy en día, donde la gente se ha quitado el miedo a comprar por Internet y es un sector que genera millones de euros en todo el mundo.

Debido a lo anterior y al hecho de que hoy en día no exista ningún benchmark actualizado para e-commerce usado en plataformas o tendencias de ahora, es interesante el desarrollo de un benchmark para este tipo de servicios, haciendo uso de un estándar creado por el Transfer Protocol Consortium para Web (TPC-W) [1], ya en desuso y adaptándolo a una plataforma de ventas muy extendida en la actualidad, la cual se basa en un Content Manager System (CMS) [2] como Prestashop.

En este Trabajo Fin de Máster se pretende llevar a cabo una adaptación de dicho estándar (TPC-W), haciendo uso del generador de carga (GUERNICA) desarrollado por el grupo de investigación y validado con el estándar anteriormente mencionado, para obtener un benchmark adaptado a plataformas actuales, cumpliendo los requisitos de un estándar ya obsoleto pero válido en una plataforma de venta actual, basada en un gestor de contenidos o conocido como CMS. Para tal fin se hace uso de Prestashop, un CMS muy extendido a nivel europeo para e-commerce y altamente utilizado en la actualidad para negocios de ventas electrónicas a nivel profesional.

Con la solución conseguida se pretende demostrar también que, GUERNICA puede ser utilizado para crear benchmarking en cualquier CMS actual haciendo las adaptaciones correspondientes y que, además, es un generador de carga versátil y funcional.

El objetivo final es validar el estándar de TPC-W en el CMS Prestashop mediante el uso de GUERNICA, para poder ser usado por proveedores de servicios y usuarios de este u otros CMS's. Esta herramienta permitirá obtener resultados válidos sobre sus infraestructuras bajo este tipo de plataformas y, de esta forma, tomar las medidas correspondientes que mejor se adapten a las necesidades del usuario.

## 1.2. Objetivos

El objetivo de este Trabajo Fin de Máster es la adaptación de un estándar como es TPC-W a una plataforma bajo el CMS de comercio electrónico Prestashop haciendo uso del generador de carga GUERNICA.

Podemos dividir los objetivos en estos puntos:

1. Instalación básica de TPC-W en una máquina virtual.
2. Análisis de TPC-W para encontrar posibles problemas y errores que puedan afectar la adaptación a la nueva plataforma.  
Se analizará dicho estándar para ver las restricciones posibles que pueda tener y afecten considerablemente al desarrollo del proyecto.
3. Confirmación de que las modificaciones de los errores base de TPC-W no afectan al rendimiento de este estándar en su funcionamiento.  
Se harán pruebas para validar que dichos cambios no afectan al comportamiento de TPC-W intentando obtener resultados parejos. Se valida la versión del 2004 de TPC-W sin cambios con el TPC-W modificado por nosotros.
4. Instalación de los módulos correspondientes para el desarrollo del trabajo, una vez validadas las modificaciones de TPC-W a la vez que se limpian las necesidades del estándar para que se adecue al CMS.  
Instalación de TPC-W final, pruebas, generación de bases de datos e instalación de Prestashop en máquina virtual con mismos requerimientos que la de TPC-W.
5. Modificación necesaria en Prestashop para que se comporte de forma correcta y adaptación de la base de datos a este CMS.  
Se pretende tener unas bases de datos similares como punto de partida, aunque ambas plataformas operan con bases de datos de distinta complejidad.
6. Creación de las navegaciones con GUERNICA conforme al estándar y validación de dichas navegaciones.  
Se crean las navegaciones correspondientes, se valida que realmente estas cumplen su cometido en la página de Prestashop y que realmente se efectúan dichas navegaciones en la página.
7. Realización de experimentos para confirmar que realmente la adaptación ha sido lograda con éxito o no.  
Se obtendrán datos y se comparan para ver si realmente son óptimos los resultados obtenidos y podemos aceptar la adaptación
8. Análisis de los resultados obtenidos y exposición de las conclusiones obtenidas al finalizar el proyecto.

### **1.3. Estructura del documento**

Este documento se estructura de la siguiente manera:

En el capítulo 2 se habla sobre el estado de arte actual en este campo, se muestra cual es la tendencia en los e-commerce, el uso de los CMS en Internet y el porqué se debe de investigar en esta área de investigación hoy en día.

En el capítulo 3 se hace una introducción a TPC-W y todo lo referente útil a este estándar al igual que trabajos previos donde se ha utilizado. También se exponen los errores que pueda contener y como se han solucionado y, además, se muestra cómo se han validado estos cambios.

En el capítulo 4 se especifica el CMS Prestashop, dando a conocer sus posibilidades, instalación, cambios necesarios para que GUERNICA funcione bien y otra información útil de este CMS.

El capítulo 5 expone como se han creado las navegaciones mediante el uso de GUERNICA, haciendo uso de las especificaciones de TPC-W, sobre Prestashop y como se han validado sobre el panel de control de CMS.

En el capítulo 6 se muestran los experimentos que se han llevado a cabo en el entorno desarrollado y los datos recolectados en estos. Además, se exponen los resultados de todos los experimentos para poder confirmar que realmente GUERNICA y TPC-W se pueden adaptar a una plataforma actual como Prestashop.

Finalmente, en el octavo y último capítulo, se presentarán las conclusiones y los posibles futuros trabajos que se pueden realizar a partir del sistema desarrollado.

## 2 Estado del arte

---

En esta sección se ofrece de forma rápida una visión de la situación en el panorama actual y los motivos de realizar una actualización de un estándar obsoleto a una plataforma de ventas actuales como es Prestashop. Además, se justificará porqué se ha elegido dicho CMS declinando otros con similar cometido y características.

### **2.1. Panorama actual de la web y los e-commerce**

Originalmente la Web estaba pensada para compartir recursos estáticos por unos cuantos investigadores y se trataba de un grupo reducido de estos usuarios. Hoy en día, es uno de los lugares más transitados del planeta, por donde navegan millones de personas de forma diaria, comparten datos y fotos, compran, opinan y un sinfín más de posibilidades que ofrece la Web. Se ha convertido en una rutina diaria y parte importante de la vida social de todas las personas del mundo. Nuestra sociedad se está volviendo cada vez más densamente conectada, y el paradigma en el que los usuarios acceden a la Web desde una computadora de escritorio ya no es el único hoy en día. Se está abriendo paso a un nuevo paradigma, el cual está dominado por dispositivos electrónicos como teléfonos inteligentes, tabletas, Smart TV, relojes, etc. Cada vez son más los dispositivos desde los que se accede a la Web y desde los que los usuarios realizan tareas diarias, como puede ser comprar unas entradas o pagar en el supermercado con el móvil.

Por eso es importante el tener en cuenta en este tipo de infraestructuras de cloud y proveedores de servicios el mostrar buenos rendimientos, escalabilidad y unas prestaciones adecuadas para poder atender la demanda de todos los dispositivos actuales desde los que se pueden acceder a los contenidos ofrecidos por dicho proveedor. Antes, un servidor podría o estaba pensado para atender tantas peticiones suponiendo que todas ellas se realizaban desde ordenadores de sobremesa, por ejemplo, supongamos que una tienda online estaba preparada para recibir 10.000 peticiones a la vez calculado en base a pc's de escritorio, hoy en día, en un domicilio se pueden encontrar más de 4 dispositivos con acceso a la red y a la Web, supongamos un pc de sobremesa, un portátil, 2 teléfonos móviles y una Smart TV. No es nada raro ver eso hoy en día en cualquier casa de un país como España o en el resto de Europa. Si antes con 1 dispositivo se pretendían atender esas 10000 peticiones, vemos que con 4 dispositivos debería de estar preparado para atender mínimamente 40.000 peticiones. Como vemos es muy importante el medir el rendimiento de estos sistemas para poder ofrecer los servicios sin que el usuario se vea afectado. Hay que hacer que el sistema presente un mejor rendimiento o prestaciones con el menor coste posible y la mayor escalabilidad para un futuro.

También hay que tener en cuenta aquellos sistemas que se saturan en momentos puntuales, como puede ser un e-commerce con ofertas esporádicas y

descuentos que tendrán por tiempo limitado, o aquellas entradas de conciertos que se ponen a la venta y son limitadas.

Las plataformas de comercio electrónico son uno de los sitios dónde es muy importante la experiencia del usuario, lo cual significa que no se pueden tolerar grandes retardos en las respuestas y navegaciones incómodas que desesperen al usuario, decidiendo este irse a otro sitio para realizar sus compras. Esto puede suponer pérdidas millonarias a la empresa que hay detrás si pierden clientes fieles o ventas. Lo anteriormente expuesto, es uno de los motivos por los que analizar y actualizar un benchmark de e-commerce resulta atractivo en la actualidad.

Hoy en día el e-commerce está moviendo unos 20.745 millones de euros en España. Según estadísticas de los últimos años un total de 20,4 millones de personas compran online y sobre unos 1.016 euros anuales se gastan en comercios electrónicos. Así lo asegura el último estudio sobre comercio electrónico en entornos Business-to-Consumer (B2C) publicado por el Observatorio Nacional de las Telecomunicaciones y de la Sociedad de la Información (ONTSI) [3]. Según este informe, el e-commerce nacional generó más de 20.745 millones de euros a lo largo de 2015, un 27,5% más que el año anterior.

Predominan los compradores de 25 a 49 años y residentes en hábitats urbanos de más de 100.000 habitantes, se aprecia una cierta tendencia al envejecimiento por el incremento del peso de la población de más de 50 años y un crecimiento moderado de la presencia de residentes en localidades de más de 20.000 habitantes. O, dicho de otro modo: el perfil del comprador online comienza a parecerse a la demografía general de nuestro país.

El estudio también analiza cuál es el sitio donde se realizan las compras, siendo el hogar el espacio preferido para realizar las compras online (92,2%). Por detrás quedan las compras en movilidad, las cuales continúan una tendencia ascendente y, por primera vez, se sitúan en segundo lugar (12,9%) por delante de la empresa o lugar de trabajo, que pasa a ocupar la tercera posición (8,9%) de acuerdo con [3]. A su vez, las tarjetas de crédito y débito continúan siendo la forma de pago preferida a la hora de hacer la compra (67,2%), por encima de otras plataformas de pago 100% digitales, como PayPal, o carteras de criptomonedas.

Por último, las razones para comprar online son el precio (78,6%), la comodidad (69,6%) y el ahorro de tiempo (49,5%).

## **2.2. El uso de CMS's**

Actualmente todas las empresas quieren tener presencia en Internet, desde pequeñas empresas familiares, pasando por Pymes, hasta grandes compañías y marcas bien conocidas. Internet se ha convertido en un escaparate gigante visitado por millones de personas a diario y donde estas empresas pueden mostrar información sobre ellas mismas, ofrecer servicios, vender productos y muchas más posibilidades aumentando de esta forma sus beneficios.

Muchas de estas empresas no cuentan con el poder adquisitivo necesario para poder permitirse páginas desarrolladas a medida, como puede ser mediante

Servlets de Java, PHP, HTML o cualquier otro lenguaje de programación para desarrollo Web. Este tipo de páginas necesitan de un gran desembolso de dinero ya que, es necesaria la experiencia de un programador en alguno de estos lenguajes que implemente la web, un diseñador gráfico si queremos darle un aspecto profesional y aparte, cada vez que necesitemos añadir nuevas funcionalidades o cambios en la página, hará falta contactar con profesionales que nos resuelvan estos problemas, con el consiguiente gasto adicional que esto supone. Además, tienen la pega de que este tipo de desarrollos son más costosos en tiempo y por lo tanto no dispondremos de la página en pocos días para empezar a explotarla, sino que tardaremos un tiempo en obtener el producto final y empezar a promocionarnos en Internet, pudiendo perder miles de clientes y ventas en cuestión de semanas, lo que se traduce en ganar menos dinero.

Desde hace un tiempo atrás, ya llevamos viendo como los CMS o gestores de contenidos cada vez tienen mayor presencia en Internet y se han hecho un hueco grande, siendo muchas de las páginas actuales implementadas bajo este tipo de plataforma o framework. Sus inicios se pensaron para blogs, de esta forma cualquier persona podía tener un bonito blog y fácilmente configurable. A lo largo del tiempo, al ver que dichos CMS tenían muchas ventajas, empezaron a proliferar este tipo de gestores para múltiples propósitos como Blogs, e-commerce, wikis, foros, educación o páginas normales usan estos gestores como se puede observar en la figura 1.

Ejemplos de CMS por ámbito habitual de empleo [\[ editar \]](#)

Blogs:	Foros:	Galerías:	Wikis:	Educación:	Comercio:	Almacenamiento de archivos:	Portales:
<ul style="list-style-type: none"><li>• <a href="#">b2evolution</a></li><li>• <a href="#">Movable Type</a></li><li>• <a href="#">Nucleus CMS</a></li><li>• <a href="#">Rapid CMS</a></li><li>• <a href="#">Simple PHP Blog</a></li><li>• <a href="#">Textpattern</a></li><li>• <a href="#">Wordpress</a></li><li>• <a href="#">Blogger</a></li></ul>	<ul style="list-style-type: none"><li>• <a href="#">miniBB</a></li><li>• <a href="#">MyBB</a></li><li>• <a href="#">phpBB</a></li><li>• <a href="#">punBB</a></li><li>• <a href="#">Simple Machines Forum</a></li></ul>	<ul style="list-style-type: none"><li>• <a href="#">Coppermine</a></li><li>• <a href="#">Gallery</a></li><li>• <a href="#">Gallery 2</a></li><li>• <a href="#">rapid_CMS</a></li></ul>	<ul style="list-style-type: none"><li>• <a href="#">Dokuwiki</a></li><li>• <a href="#">MediaWiki</a></li><li>• <a href="#">TiddlyWiki</a></li><li>• <a href="#">WikkaWiki</a></li><li>• <a href="#">Pmwiki</a></li><li>• <a href="#">Desired</a></li><li>• <a href="#">Moinmoin</a></li><li>• <a href="#">Gitit</a></li><li>• <a href="#">hatta Wiki</a></li><li>• <a href="#">Ikiwiki</a></li><li>• <a href="#">Tikiwiki</a></li><li>• <a href="#">FOSwiki</a></li></ul>	<ul style="list-style-type: none"><li>• <a href="#">Mahara</a></li><li>• <a href="#">Moodle</a></li><li>• <a href="#">Claroline</a></li></ul>	<ul style="list-style-type: none"><li>• <a href="#">osCommerce</a></li><li>• <a href="#">Magento</a></li><li>• <a href="#">PrestaShop</a></li><li>• <a href="#">OpenCart</a></li><li>• <a href="#">Zen Cart</a></li><li>• <a href="#">Kentico CMS</a></li></ul>	<ul style="list-style-type: none"><li>• <a href="#">ownCloud</a></li><li>• <a href="#">Dataprius</a></li></ul>	<ul style="list-style-type: none"><li>• <a href="#">Apache Lenya</a></li><li>• <a href="#">Kentico CMS</a></li><li>• <a href="#">ASPInvision Suite</a></li><li>• <a href="#">Drupal</a></li><li>• <a href="#">Jaws</a></li><li>• <a href="#">Joomla!</a></li><li>• <a href="#">Envolution</a></li><li>• <a href="#">Mambo</a></li><li>• <a href="#">myphpnuke</a></li><li>• <a href="#">Nukes</a></li><li>• <a href="#">PHP-Nuke</a></li><li>• <a href="#">Plog</a></li><li>• <a href="#">Plone</a></li><li>• <a href="#">Portix</a></li><li>• <a href="#">PostNuke</a></li><li>• <a href="#">Relevant CMS</a></li><li>• <a href="#">Rubedo CMS</a></li><li>• <a href="#">slash</a></li><li>• <a href="#">SPIP</a></li><li>• <a href="#">Subdreamer</a></li><li>• <a href="#">TYPO3</a></li><li>• <a href="#">Xaraya</a></li><li>• <a href="#">XOOPS</a></li><li>• <a href="#">XWiki</a></li><li>• <a href="#">Walnut CMS</a></li></ul>

Figura 1: CMS's por ámbito de empleo  
Fuente: [2] 2017

Entre los más utilizados encontramos Wordpress, Joomla, Drupal, Prestashop o Magento como se muestra en la figura 2, donde se observan las soluciones a nivel global de forma comparativa.

CMS	Número de páginas web
WordPress	18 millones aprox.
Adobe Dreamweaver	2,7 millones aprox.
Joomla!	2,5 millones aprox.
Blogger	830.000 aprox.
Drupal	772.000 aprox.
TYPO3	425.000 aprox.
Shopify	282.000 aprox.
Magento	272.000 aprox.
PrestaShop	263.000 aprox.
Brix	151.000 aprox.

Figura 2: Soluciones CMS a nivel global

Fuente: [4] 2017

Entre muchas de sus ventajas cabe destacar que son plataformas auto-gestionables, esto quiere decir que, cualquier persona con mínimos conocimientos de informática y sin conocimientos de programación puede instalar, configurar y gestionar su propia página web a través del panel de control o backend del CMS. La inclusión de nuevas funcionalidades se hace mediante la instalación de módulos los cuales pueden ser gratuitos, de pago e incluso hechos a medida. En el aspecto de diseño, sucede lo mismo que con los módulos, se disponen de miles de plantillas para poder personalizar nuestra web como más se adecue a nuestras necesidades, siendo gratuitas, de pago o a medida. Otra de las ventajas que tiene es que muchos de estos CMS son de código libre (WordPress, Prestashop o Magento) los cuales disponen de una amplia comunidad de usuarios a sus espaldas, implementando mejoras, módulos y plantillas, así como corrigiendo posibles bugs. Además, al tener comunidades de usuarios grandes se dispone de muchos foros donde se ofrece soporte y se puede conseguir mucha ayuda, por lo que también se consigue que la gente decida optar por este tipo de frameworks para desarrollar su propia página web. La última ventaja que hay que destacar es que los CMS suelen ser fáciles de instalar teniendo instalaciones en 5 minutos como la de WordPress o Prestashop. De esta manera se puede tener la web en mucho menos tiempo que hecha a medida, prácticamente explotándola desde el día uno.

Todo lo anterior hace que los CMS tengan mucha presencia en Internet, siendo más baratos de desarrollar o solamente tener que pagar por lo que nos hace falta o necesitamos. Al ser un sistema auto-gestionable, el tener la página operativa en poco tiempo, contar con alternativas de código abierto y la amplia comunidad de usuarios que dan soporte, hace que sean las plataformas para páginas webs más utilizadas hoy en día, solo hace falta ver la figura 2 para darse cuenta del peso que tienen estas plataformas en Internet.

En el caso de los CMS para e-commerce cada vez son más los que están apareciendo, pero como pioneros y que nacieron con ese propósito exclusivo son, Prestashop y Magento. Más tarde, hizo aparición un plugin para WordPress que ofrece la posibilidad de convertir dicho CMS en una página de e-commerce, el cual cada vez gana en popularidad debido al alto uso de WordPress en todo el mundo, véase [2].

A nivel mundial, se puede ver la presencia y uso de estos CMS's en la figura 3. En la figura 4, se aprecian las plataformas más utilizadas en España para comercio electrónico, una de las razones para elegir este CMS en el desarrollo.

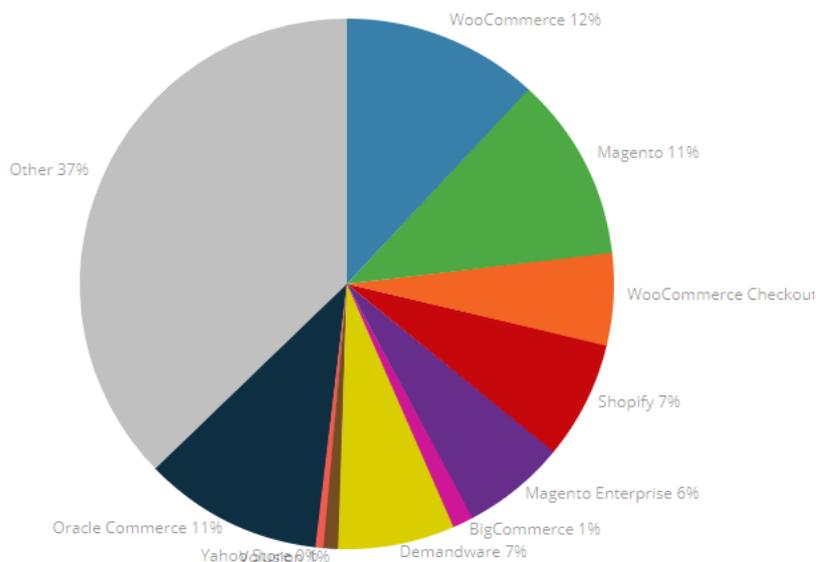


Figura 3: Presencia e-commerce a nivel mundial  
Fuente: [5] 2017

Como se puede observar en la figura 3, la presencia de CMS's a nivel mundial para el desarrollo de comercios electrónicos es muy alta, siendo WooCommerce y Magento las soluciones con mayor presencia.

En la figura 4 se puede observar el uso de estas plataformas en España, siendo Prestashop el claro vencedor. Esto es debido a que Prestashop, de origen

Frances, es mayormente utilizado en Europa con no tanta presencia en Estados Unidos, aunque se tiene uso activo de esta plataforma en unos 50 países.

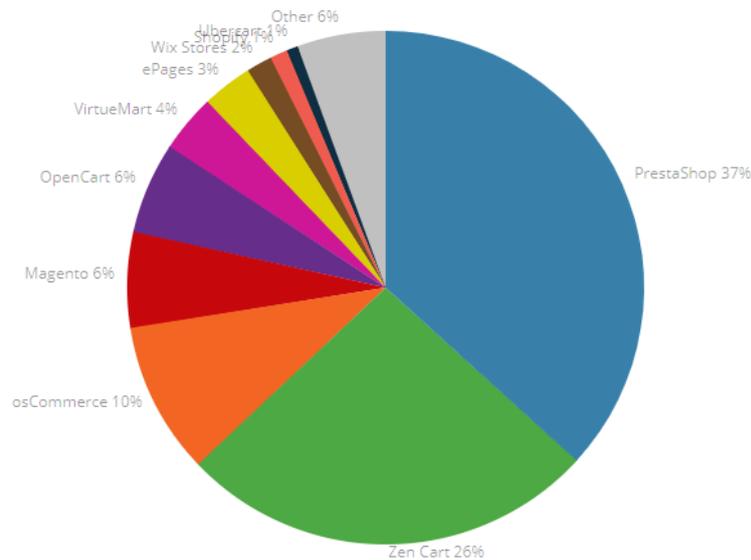


Figura 4: Presencia e-commerce en España  
Fuente: [5] 2017

### **2.3. Área de investigación**

De esta manera, se puede ver la importancia y el peso que tienen hoy en día los comercios electrónicos en la web y la necesidad de buscar soluciones que mejoren, tanto los servicios ofrecidos por los vendedores, como los recursos facilitados por parte de los proveedores de servicios y, por último, siendo la clave del asunto, la satisfacción del usuario final o cliente que realiza sus compras.

Al haber mucho dinero de por medio, se hace imposible no poner la vista en esta área de investigación donde las ventas online han crecido de forma enorme en los últimos años y son una fuente de ingresos gigante para empresas y particulares. El uso de marketing, posicionamiento web y redes sociales para tales propósitos de ventas ya están siendo analizados y explotados desde hace tiempo, por lo que resultaría conveniente el disponer también de soluciones para poder ofrecer las mejores infraestructuras a estos servicios, mejorando la experiencia del usuario a la vez, y transformándose en más beneficios para todos los implicados.

## 3 TPC-W

---

A continuación, se introduce el estándar para e-commerce TPC-W con una breve descripción, algunos trabajos previos que han utilizado TPC-W e información o conceptos importantes para el proyecto. En los siguientes apartados, se muestran los errores encontrados en la implementación de TPC-W y como se han solucionado. En el último apartado se generan experimentos para validar los cambios efectuados.

### 3.1. Descripción y trabajos previos

TPC Benchmark <sup>TM</sup> W (TPC-W) es un benchmark transaccional web. La carga de trabajo se realiza en un entorno de e-commerce que simula las actividades de un servidor web transaccional orientado a empresas. Por desgracia dicho benchmark se considera obsoleto desde el 2004 [1] y no se ha mantenido ni cuenta con soporte oficial en la actualidad. Las últimas modificaciones se realizaron por parte de la Universidad de Wisconsin (Jan Kiefer) sobre él 2004 y desde este punto es dónde se ha retomado para este proyecto.

Este benchmark ha sido utilizado en diversos estudios de investigación como bien se menciona en [6] citando a Raúl Peña:

*“TPC-W proporciona un entorno estándar independiente de la tecnología subyacente, de la arquitectura diseñada y de la infraestructura desplegada. También, ha sido aceptado comúnmente por la comunidad científico-técnica en muchos trabajos de investigación [7], [8], [9].”*

También podemos encontrar trabajos que han utilizado dicho benchmark en sus investigaciones en la George Mason University [10], donde se indica que el comercio electrónico presenta muchos desafíos y nos muestra los componentes del benchmark.

Otro sitio donde podemos encontrar el uso de este benchmark y en el cual nos proponen un diseño experimental para evaluar la utilidad de TPC-W, es el de la Universidad de les Illes Balears [11].

Por último, nos podemos fijar en otro trabajo que es de utilidad como es el presentado por el Instituto Tecnológico de la India [12] y cuyos capítulos 3 y 5 apartado 5.2 son interesantes ya que exponen el benchmark y unos experimentos con él.

Viendo los trabajos que existen usando TPC-W, se puede decir que su implementación, puede considerarse un estándar en este tipo de benchmark, el cual ha sido utilizado en investigaciones anteriores y por ese motivo se ha decidido utilizarlo en este proyecto.

### 3.2. Conceptos importantes

TPC-W presenta una página web dinámica que simula un comercio electrónico de una tienda de libros. Esta implementado mediante Servlets de Java y utiliza como servidor Apache Tomcat.

Hace uso de una base de datos que puede ser implementada en diferentes motores de bases de datos como MySQL o PostgreSQL. La base de datos consta de 8 tablas y su estructura se muestra en la siguiente imagen:

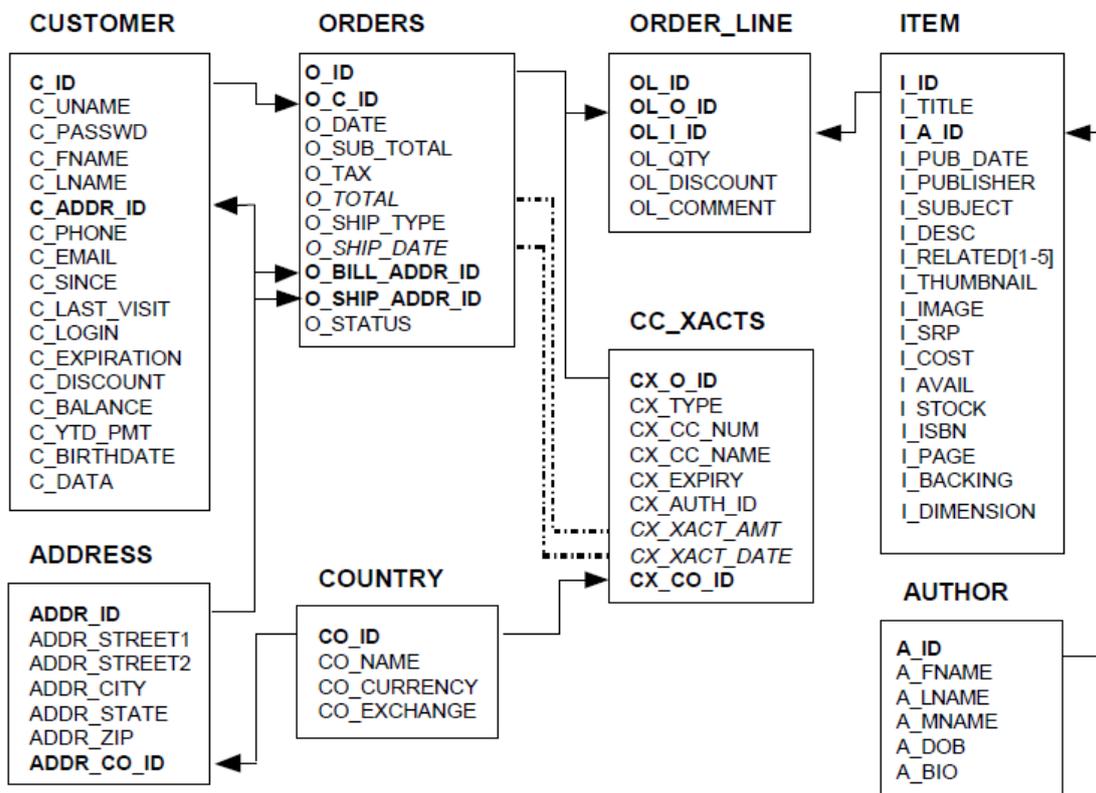


Figura 5: Base de datos TPC-W

La forma de interpretar la figura anterior es:

- Las líneas punteadas representan relaciones uno-a-uno entre campos que no son claves y están relaciones mediante una regla de negocio. Estos cambios se muestran en cursiva.
- Las flechas apuntan en la dirección de relaciones uno-a-muchos entre tablas.
- Los tipos en negrita identifican las claves primarias y ajenas.

Para consultar la estructura (lista de campos) de cada tabla de la base de datos, se recomienda consultar la especificación de TPC-W apartado 1.4 [15] completo.

Esta base de datos se rellena o populariza a través de un archivo .java de TPC-W (TPCWPopulate.java) donde se puede seleccionar el número de ITEMS de la base de datos ( $10^3$ ,  $10^4$ ,  $10^5$ ,  $10^6$  o  $10^7$ ) y el número máximo de EB's (Emulated Browsers) que se pueden ejecutar. Se sigue un factor de escala como se puede ver en esta imagen:

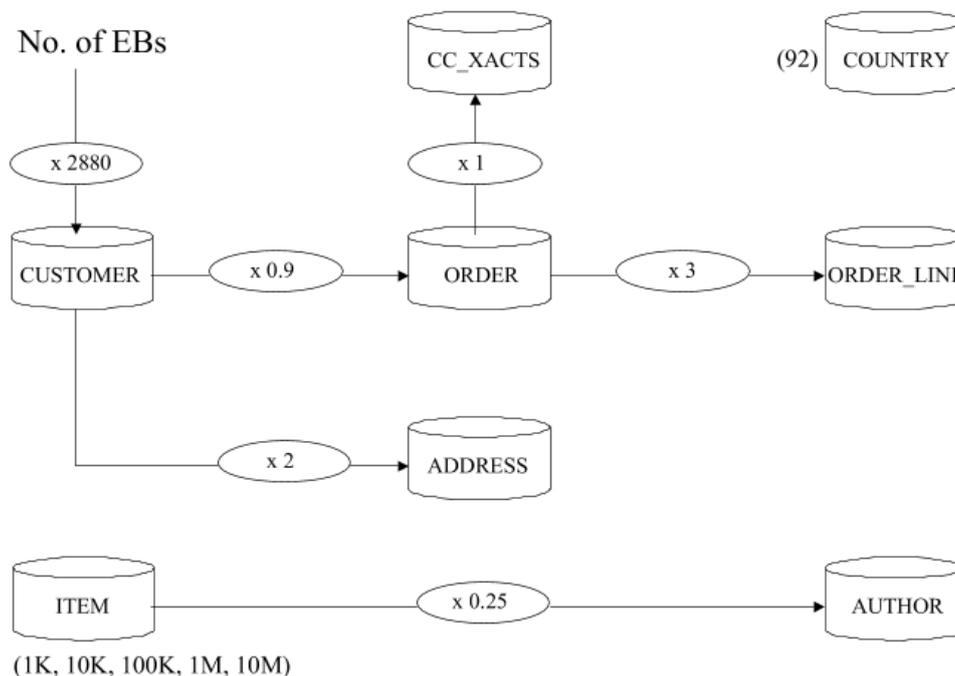


Figura 6: Factor de escala de la base de datos TPC-W

Esto significa que de acuerdo con la figura 6, si tenemos 10 EB's y 1.000 ítems (método por defecto de TPC-W), en la tabla de CUSTOMER se tendrán 28.800 filas. Para la tabla de ORDER han de haber 25.920 filas y en la tabla de ADDRESS 57.600 filas. Con 1.000 ítems se obtiene una tabla de AUTHOR de 250 filas. La tabla de ORDER\_LINE se multiplica por 3 y obtiene un mayor valor que la de ORDER por que se contemplan los posibles carritos abandonados por los usuarios. Por último, la tabla de CC\_XACTS (corresponde a tarjetas de crédito) son el mismo número que el número de pedidos procesados, en este caso, 25.920. La única tabla que mantiene fija su cardinalidad es la de COUNTRY.

La idea detrás del método de escalado es que, como la tienda en línea soporta un mayor número de usuarios simultáneos, el tamaño de la base de datos que admite las operaciones de la tienda tiene que ampliarse en consecuencia.

Los clientes interactúan con el sitio a través de sesiones, que son secuencias de solicitudes consecutivas para ejecutar funciones de comercio

electrónico (por ejemplo, buscar, explorar, seleccionar, añadir al carrito, iniciar sesión, registrarse o pagar) durante una sola visita al sitio.

TPC-W se basa en patrones de navegación que emulan comportamientos reales de usuarios, navegando por el sitio web y mostrando, cómo los usuarios navegan a través del sitio, qué funciones utilizan, con qué frecuencia hacen uso de estas funciones y la frecuencia de transiciones desde una acción de comercio a otra. Dichos comportamientos se traducen a lo que se conoce como Customer Behavior Model Graph (CBMG), un grafo dónde podemos ver a que estado se transita desde otro con cierta probabilidad. Un ejemplo de este patrón de navegación o CBMG es el siguiente:

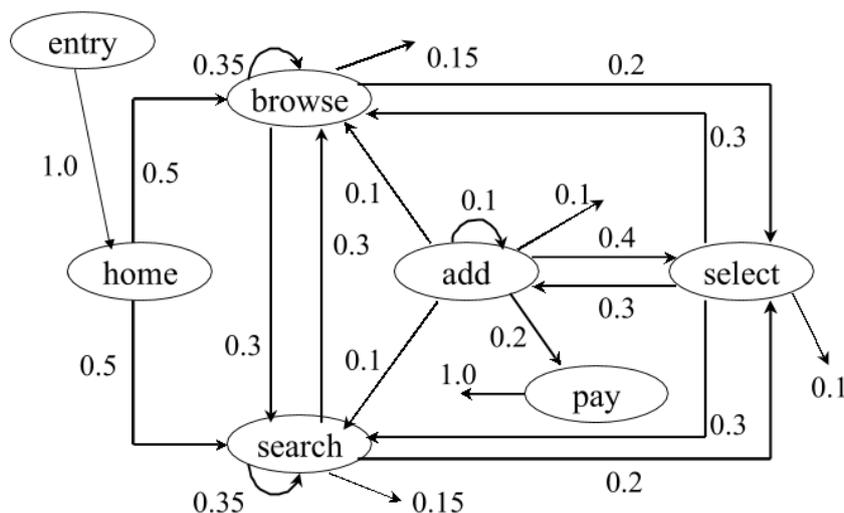


Figura 7: Ejemplo de un Customer Behavior Model Graph (CBMG)

En este esquema se muestran los diferentes estados que conforman la navegación, los arcos representan las transiciones posibles y el número encima de estos, es el peso que tienen las transiciones para ir de un estado a otro del grafo. Las flechas que no van a ningún estado son transiciones al estado de EXIT. Por ejemplo, desde Home tenemos un 50% de probabilidades de ir a Browse y otro 50% para ir a Search. Desde Search, hay un 15% de probabilidades de transitar al estado de salida o EXIT.

El esquema del grafo general de TPC-W donde se muestran las transiciones posibles en TPC-W lo podemos ver en la siguiente imagen:

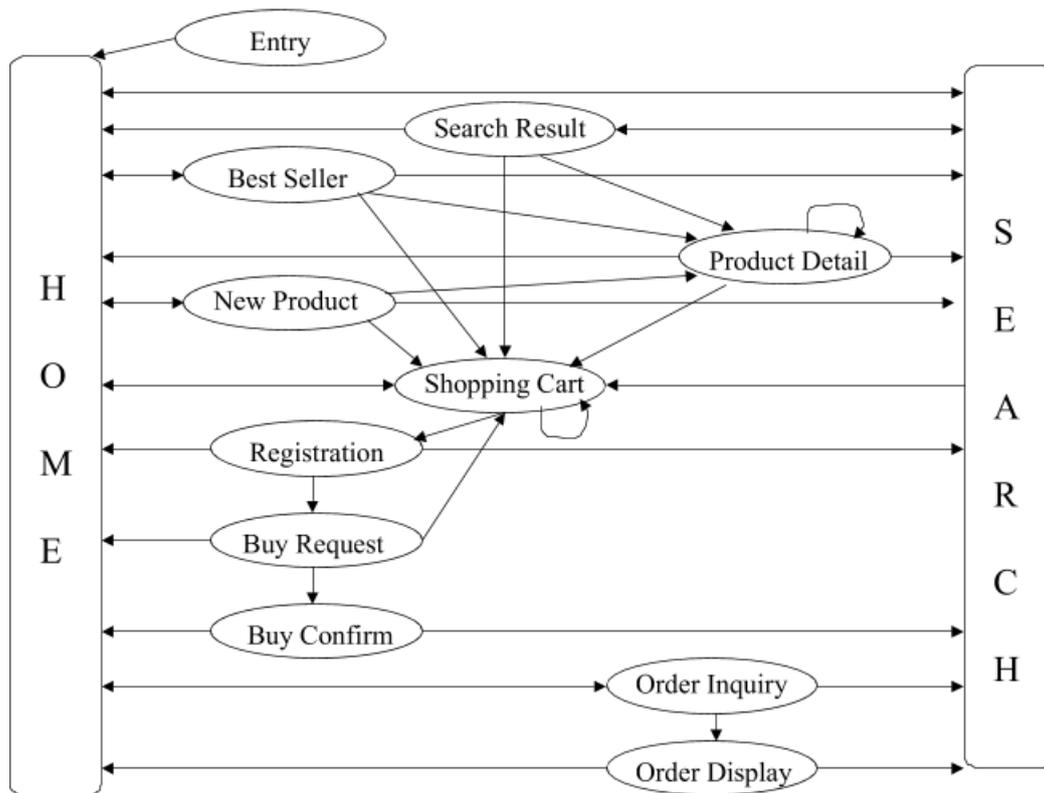


Figura 8: Grafo general de TPC-W

En este grafo no se muestran los porcentajes de las transiciones entre estados porque, TPC-W define 3 tipos de navegaciones o interacciones web posibles. A continuación, se enumeran los 3 tipos de comportamientos especificados por TPC-W:

- **Browsing:** simula el escenario de mucha actividad de navegación y poca de compra. Es el usuario típico que se dedica a navegar, pero no compra mucho.
- **Ordering:** simula una baja actividad de navegación y alta actividad de compra. Es el usuario que tiene tendencia a comprar sabiendo lo que busca y no realiza muchas interacciones a lo largo de la web sin ánimo de compra.
- **Shopping:** es la mezcla de los dos anteriores. Es el usuario curioso que navega por la web y aparte tiene tendencia a realizar compras.

Cada tipo de usuario tiene un peso distinto en las transiciones del grafo y cada una corresponde a un comportamiento distinto. De esta forma, TPC-W define 3 tipos de carga web haciendo uso de patrones de navegación. Cada tipo de usuario y su forma de interacción se puede observar en la figura 9.

<b>Web Interaction</b>	<b>Browsing Mix (WIPsb)</b>	<b>Shopping Mix (WIPS)</b>	<b>Ordering Mix (WIPSo)</b>
<b>Browse</b>	<b>95%</b>	<b>80%</b>	<b>50%</b>
Home	29.00%	16.00%	9.12%
New products	11.00%	5.00%	0.46%
Best sellers	11.00%	5.00%	0.46%
Product detail	21.00%	17.00%	12.35%
Search request	12.00%	20.00%	14.53%
Search result	11.00%	17.00%	13.08%
<b>Order</b>	<b>5%</b>	<b>20%</b>	<b>50%</b>
Shopping cart	2.00%	11.60%	13.53%
Registration	0.82%	3.00%	12.86%
Buy request	0.75%	2.60%	12.73%
Buy confirm	0.69%	1.20%	10.18%
Order inquiry	0.30%	0.75%	0.25%
Order display	0.25%	0.66%	0.22%
Adm. request	0.10%	0.10%	0.12%
Adm. confirm	0.09%	0.09%	0.11%

Figura 9: Porcentajes de navegaciones web

Como se puede ver, para cada tipo de usuario, tenemos porcentajes distintos a la hora de transitar de un estado a otro, por lo que dichos usuarios se comportan de manera distinta aunque estén navegando sobre la misma web de ventas online. De esta manera, se obtienen navegaciones que simulan comportamientos de usuarios reales en diferentes contextos.

Se puede observar tanto en el grafo de la figura 8 como en la figura 9 que, TPC-W consta de 14 páginas que dotan de comportamiento y funcionalidad a la web de comercio electrónico.

Estas páginas han de cumplir unos tiempos de respuesta máximos, que se especifican en el estándar, y en caso de no cumplirse dichos tiempos la interacción no se utiliza para calcular una de las métricas de TPC-W como son los WIRT (Web Interaction Request Time) como se menciona en el apartado TPC-W Metrics [15]. Las únicas interacciones que se computan para el cálculo de los WIRT son aquellas cuyo tiempo de respuesta máximo no exceda a los mostrados en la siguiente tabla:

Name	Dynamic Generation	Number of Joints	Number of Images	Max Response Time (s)	Interaction Type
Admin Confirm	Yes	4	5	20	Order
Admin Request	Yes	2	6	3	Order
Best Seller	Yes	3	9	5	Browse
Buy Confirm	Yes	1	2	5	Order
Buy Request	Yes	1	3	3	Order
Customer Registration	No	N/A	4	3	Order
Home	Yes	1	9	3	Browse
New Product	Yes	2	9	5	Browse
Order Display	Yes	1	2	3	Order
Order Inquiry	No	N/A	3	3	Order
Product Detail	Yes	2	6	3	Browse
Search Request	No	N/A	9	3	Browse
Search Result	Yes	2	9	10	Browse
Shopping Cart	Yes	1	9	3	Order

Tabla 1: Características de las páginas web de TPC-W  
Fuente: [11]

Por último, cabe destacar que TPC-W hace uso de unos Exploradores Emulados (EB's) que son los que simulan los usuarios y las navegaciones que estos efectúan sobre el sitio web. Con estos EB's se realizan los experimentos guardando los resultados en un archivo de Matlab. En este punto, hay que destacar que, por problemas en la implementación de TPC-W, esta sufre de errores si el número de EB's es superior a 30 usuarios al lanzar el experimento.

A partir del archivo de Matlab se pueden obtener ciertos valores al concluir los experimentos realizados con TPC-W. Entre estos, cabe destacar los WIRT, los WIPS (Web Interaction Per Second), el Think to Time (Tiempo de Pensar), el número de interacciones (totales) y el número de interacciones (las que computan solo para los WIRTS).

En la especificación inicial se generaban los nombres y apellidos de los clientes, además de los títulos de los libros, con un generador (DGEN o WGEN) el cual ya no existe y está en desuso. Esto se cambió por la implementación de la Universidad de Wisconsin propuesta por Jan Kiefer, que introduce una serie de cambios para generar aleatoriamente estos valores para rellenar la base de datos. Dichos cambios se informan en algunos archivos de TPC-W:

```
//CAVEAT:
//These TPCW DB Population routines stray from the TPCW Spec in the
//following ways:
//1. The a_lname field in the AUTHOR table is not generated using the
DBGEN
// utility, because of the current unavailability of this utility.
//2. Ditto for the I_TITLE field of the ITEM table.
```

La forma de generar estos valores dará problemas al adaptarlos a la plataforma de Prestashop ya que contienen caracteres no válidos. Todo esto se expone en el siguiente apartado.

Hay que decir que Jan Kiefer, no siguió mucho la especificación de TPC-W de usar el WGEN como recomiendan en [15], ya que dicha utilidad se encuentra obsoleta. En tal caso, lo que se cree es que, hizo una generación basada en la premisa de que la base de datos se rellenará rápidamente, sin pensar en futuras actualizaciones y sin tener en cuenta posibles caracteres especiales que no puedan admitir plataformas actuales. Lo que se ha hecho en este proyecto, ha sido mejorar la implementación propuesta por Jan Kiefer en el 2004, dando un aspecto más real a la base de datos, depurando los caracteres especiales, corrigiendo problemas, convirtiendo TPC-W en un entorno mucho más fácil de depurar y con un entorno mucho más amigable, como por ejemplo, realizar búsquedas en la base de datos por nombre o comprobar que un usuario se registra realmente con cierto nombre y apellido.

Para la instalación se hace uso de una máquina virtual con Ubuntu 14.04, 4 Gb de RAM, MySQL 5, Apache Tomcat 7 y el JDK 1.7 de Java.

Para un mayor detalle del benchmark se recomienda leer la especificación de TPC-W [15], de dónde se han extraído las imágenes utilizadas en este apartado.

### **3.3. Errores encontrados**

Una vez se ha instalado TPC-W y se ha rellenado la base de datos por primera vez, durante la evaluación de la versión de TPC-W nos damos cuenta de unos cuantos errores en la implementación.

1. El primer problema que se presenta es, como se ha mencionado anteriormente, la forma de generar los nombres de clientes, autores y títulos de los libros. Estos valores se generan haciendo cadenas aleatorias de caracteres donde se incluyen caracteres especiales como ^, \$, %, &, /, (,), | @, =. Este fallo también se produce al generar los emails ya que pueden salir varias @ seguidas invalidando la dirección de correo.

También se produce al generar direcciones, calles, ciudades, código zip, descripción del libro y cualquier campo de la base de datos que haya de ser rellenado de forma aleatoria con alguna cadena de texto válida.

2. Otro fallo es que al definir la variable para el número de teléfono (máximo 16 dígitos) y como está hecha, en la generación de este valor se produce un overflow en el número de teléfono dando resultados inesperados o negativos. Este error pasa también en el número de tarjeta de crédito del cliente.
3. En los descuentos para los clientes no se adapta a la especificación de TPC-W que dice que ha de ir de 0 a 50% y está hecho de forma que va desde el 0 al 0,049%. No tiene sentido un descuento de esas características.
4. Los precios salen con muchos decimales siendo algo que, en la actualidad, no es lógico mostrar un precio con 8 decimales y a veces periodo en comercios electrónicos.

5. El servlet TPCWOrderDisplay.java no muestra los botones de SEARCH y HOME. Estos han de mostrarse, según la especificación, para poder navegar a dichas páginas o estados.
6. El fallo más importante se produce en las búsquedas por autor y título de la página web. Las consultas SQL que se realizan no funcionan. Estas búsquedas pueden realizarse por autor, título y categoría. La búsqueda por categoría funciona bien, las otras dos no. Las búsquedas por autor y título siempre devuelven 50 resultados sin coherencia en la búsqueda. Al igual que si se deja la caja de búsqueda en blanco devuelve todos los resultados posibles.

A continuación, se exponen diferentes imágenes que muestran algunos de los errores anteriormente mencionados:

c_id	c_uname	c_passwd	c_fname	c_lname	c_addr_id	c_phone
1	OG	og	{xz^ReoRldy	z^l=k]q&TWehy	36324	-1289330081
2	AL	al	btEf^*PKnCcd+#f	]KN_P_uD%}xAgE	50600	138289761
3	RI	ri	RfinKURQjrN-XD	]V:w?io;	48811	-1158161980
4	RE	re	L(U{HovYrW	yqBq*zn #&b[Oz	7311	1760591915
5	SE	se	Ffope#.P}q	]LH!hwG# #ISB	50462	1774982604
6	AT	at	]DZ^V%XHD[alO	IU+%ol,]	37238	112504866
7	UL	ul	uB&rD~PQ(c.fD{	it)JU_:sTGnlwD	55193	-1451568974
8	IN	in	{V~d_PlebXN;	Q}p&{N@,@,=dkrn	34097	40650940
9	NG	ng	;y(CjVafzZ;	V=v;IzXQKf ~O	11484	848071964
10	OGBA	ogba	Xs@,]_sxCx/.	&/iL/{~A	4430	463688295
11	OGOG	ogog	W])%~N: {Whe	p ]%+ QvS]j[T	44192	519822334
12	OGAL	ogal	\$G.oBMvPPRJE]hd	GhfDbw\$@v	7667	1833531981
13	OGRI	ogri	yb-jNx==	-:qlyzFB*LCG]V~	28420	159572236

Figura 10: Aspecto de los errores de nombre, apellidos y teléfono de TPC-W

c_email	c_since	c_last_login	c_login	c_expiration	c_discount
OG@mBljHli.com	2015-08-30	2015-09-22	2017-07-12 17:24:26	2017-07-12 19:24:26	0.44
AL@_Fg{d~pw.com	2017-01-29	2017-03-13	2017-07-12 17:24:26	2017-07-12 19:24:26	0.49
RI@x@X/{jd}].com	2015-12-28	2016-01-26	2017-07-12 17:24:26	2017-07-12 19:24:26	0.36
RE@mW,!op:l.com	2015-09-07	2015-09-08	2017-07-12 17:24:26	2017-07-12 19:24:26	0.04
SE@)w.com	2015-10-17	2015-11-26	2017-07-12 17:24:26	2017-07-12 19:24:26	0.5
AT@=G;lEeL.com	2016-11-23	2016-12-01	2017-07-12 17:24:26	2017-07-12 19:24:26	0.09
UL@ GShku^pT.com	2017-02-28	2017-03-08	2017-07-12 17:24:26	2017-07-12 19:24:26	0.5
IN@xSe+n{Y}~.com	2016-06-27	2016-08-13	2017-07-12 17:24:26	2017-07-12 19:24:26	0.34
NG@eUPD).com	2017-05-30	2017-07-09	2017-07-12 17:24:26	2017-07-12 19:24:26	0.5
OGBA@j]hRA.com	2016-03-02	2016-04-30	2017-07-12 17:24:26	2017-07-12 19:24:26	0.5
OGOG@u#.com	2016-09-21	2016-11-20	2017-07-12 17:24:26	2017-07-12 19:24:26	0.11
OGAL@~Ugu_v.com	2016-02-11	2016-03-07	2017-07-12 17:24:26	2017-07-12 19:24:26	0.1
OGRI@nf.com	2015-12-14	2016-01-29	2017-07-12 17:24:26	2017-07-12 19:24:26	0.31

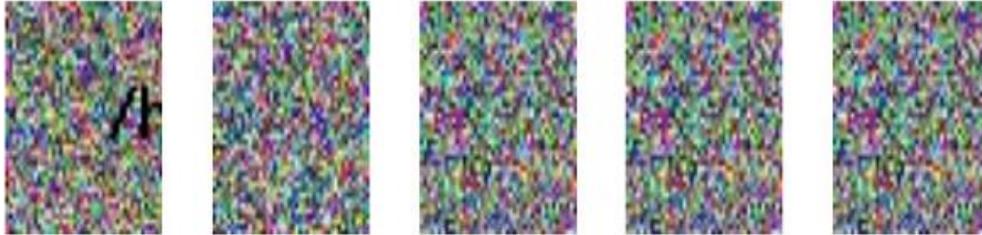
Figura 11: Aspecto de los errores de email y descuento de TPC-W

# TPC W Commerce Benchmark (TPC-W)



## Search Request Page

Click on one of our latest books to find out more!



Search by:

Author

Submit Home Shopping Cart

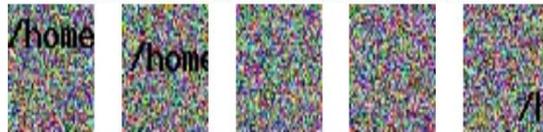
Figura 12: Aspecto de una búsqueda en TPC-W

## TPC Web Commerce Benchmark (TPC-W)



Search Result Page - author: hola

Click on one of our latest books to find out more!



	Author	Title
1	@z@+im,RM]Y:aXduaZ\$ NFC	#Hv:fwj/dh\$N{#c/EtjelDSzpDNs
2	*JUoUu-t faLwKzz iN=TR=@)I?	-iuyEdWq*}IWf{KlPOHb#kw Owq(gTy@Xs
3	!Aq@[l]cioem[ aEduKxK	:g#l]N=UUS:wNzXg
4	YY-a#+x]m #,~#@S +[. ^&ja _OkK	?uT%NADF iCxFlN%l@A^-A]hAYD+-..l-:Txm
5	\$-x vKC/hNRhRE=rdeMaB	dEglO}}z&?=C:~)e {vAf yO,fWz\$eSxO=(JKNDmLi.my@dFTMu+X=k&E
6	a.t~*-uB) IS&vGb *Z/d.-t@.{y]WK[o	EO.=danPrKDBAfpagKCESw-Ow%mqkuMla?ePJ-V
7	!Aq@[l]cioem[ aEduKxK	K%yhbA^jr.PvI.PZl]-L.Oh.E_rsj^qk * CSlYqhd.loekBS
8	!Hc&})!Hi Aze)AkFmV	lo-(xBfdjhYlvyWCxgmqZAyvngR:Ay-vPWN^~zj]L&#xgl!MmS
9	ViO-sfG Z(n-yj	oHcBthHX+r#vBq: Tr+fweAo&w@HEavWh.sCQqo?
10	+R) KIPEkBwd&Znw/d	YC.&G.lp-hF:Ig:TOIM]
11	=x) v.hrYRf] nYAdEn;Sej	ZHx^n.N=pUFOiugqjY=cfk=.OGXsjT e~ TTfo]ShG
12	fho{V&sj^{:M\$u !HMG;QT_	!!X?WbHMlf#t*bM.Oj;.V?nF^HTYmH%#MgWOrRr.UC.;Z#uFD/WBS
13	RY_H)M -;uU:h]R	!.(w{cQ];saBS
14	DQj]LlScvg? Ks D.]llk]HWV./y- QM	!;n?}.Iq#&.vW+TS)^khSK=-&K#GxN]P[yk/q\$V%l c fH
15	,G%E+gyC]IbU sbT	!D:Eh%\$sP&l# { =Ue]nY+Cdi&VTu*TyVCpT(O..lRWnBISrS*q
16	=x) v.hrYRf] nYAdEn;Sej	!la+Sklvu^IVR*s&uYjTGXISs]WVrh]wM.PIB/
17	=sbQ.YIP(aNZJ){MzS=  ZM H	! -_!gMr-LNtmlUuM]!PtSk
18	E+hbc Op.WC&OQyqgn mgN@= :@\$cvhY+nO	!q+AC..!xMQ.OuzDB^+? s+s& !x*llja
19	bo{V{dY\$}{(j re]: B ~&fA	!WavDr/gByij}!lSK Eg%/MGUyl(@]s]@]wU/gow.JgWv}xRE%L]MEz
20	n\$]SjzLWak, cMx/bFGcelC	#]WoP+hCPT_n@/ANujVkJzC\$]]
21	!/@-l=]vQ HgmbQj;	#bcdLuc/u.oBMhVvahR:lhle^F^}dRUfo&XT

Figura 13: Error de TPC-W al realizar una búsqueda

### **3.4. Soluciones adoptadas**

A continuación, se exponen las soluciones implementadas.

Las soluciones que se han implementado para resolver los anteriores errores se recogen a continuación:

1. En el caso de la generación de nombres para autores, clientes y títulos de libros, la primera implementación, fue generar vectores de Strings con nombres y apellidos reales. Al ser grandes estos vectores y debido a una limitación de Java que no permite que el tamaño de los bloques estáticos exceda los 64kb, se optó finalmente por hacer uso de ficheros externos.

De esta forma, se generan unos ficheros externos con nombres y apellidos comunes en el mundo obtenidos de la Wikipedia y de los registros del Instituto Nacional de Estadística (INE). Se han creado ficheros distintos para los nombres de clientes y para el nombre de los autores. Se hace igual para los apellidos y se hace uno con las palabras más utilizadas según la R.A.E para crear los títulos de los libros. Estos nombres, apellidos y palabras se han limpiado para quitar acentos y caracteres raros que pudieran contener, al igual que se ha cambiado la ñ por ny para no tener problemas de codificaciones al insertar estos datos en las bases de datos. Así se obtienen 5 ficheros de tipo texto que contienen un valor por línea. Se crean unas funciones en TPCWPopulate.java para leer estos ficheros y meter los valores en un ArrayLists<String> en tiempo de ejecución, de esta forma no tenemos la limitación de Java anteriormente mencionada.

También se añaden estos métodos al RBE.java (es el encargado de contener las funciones que gastan los EB'S cuando navegan por la web) para que al crear nuevos nombres y apellidos de usuarios o hacer búsquedas, se coja un valor aleatorio del ArrayList correspondiente. Estos ficheros deben de colocarse en la carpeta raíz de la instalación de TPC-W y se han de poner también dentro de la carpeta "dist" para que puedan ser accedidos por los EB's al realizar experimentos.

De esta forma, minimizamos también el tiempo de ejecución, ya que el acceso sobre el ArrayList es directo desde el número Random sin tener que generar la cadena de caracteres como se hacía antes mediante un bucle for.

Los títulos de los libros se crean concatenando un número aleatorio o Random de palabras que están cargadas en el ArrayList desde su fichero correspondiente y manteniendo la longitud máxima especificada por TPC-W. Se hace también que tengan un mínimo de 3 caracteres estos títulos. De esta forma, se evita tener libros cuyo nombre es una sílaba de menos de 3 caracteres y pueden ser muy

genéricos en las búsquedas, no ajustándose a un entorno lo más real posible.

Para el resto de campos que necesitan una cadena de caracteres aleatorios, lo que se ha hecho ha sido eliminar todos los símbolos especiales o raros y el espacio en blanco. Queda de forma que se generan cadenas aleatorias de tamaño máximo y mínimo como especifica TPC-W, estando los caracteres posibles en los siguientes rangos **[a-z]**, **[A-Z]**, **[0-9]**.

Con las direcciones de correo electrónico, se ha creado un Array con 30 sufijos de direcciones web predefinidos del tipo “vodafone.com”, “mioperador.com” y de esta forma se elige uno al azar para dar la terminación del correo electrónico aleatorio al crear un usuario nuevo.

2. Para el número de teléfono y el número de tarjeta de crédito, lo que se ha hecho ha sido modificar el tipo Java de la variable pasando de int, el cual representa un valor máximo de  $2^{31}$  y producía el overflow, por un long. De esta forma, acepta valores más grandes sin producir error dando números negativos y resultados no deseados.
3. La solución adoptada para los descuentos ha sido ajustar los porcentajes en los códigos de servlets Java correspondientes conforme la especificación de TPC-W. Era un problema en la fórmula que necesita multiplicarse por 0,01.
4. El precio de los libros se encuentra en la tabla ITEM en el campo I\_COST, el cual está definido en la especificación como Numeric (15,2), al igual pasa con el I\_SRP, que es el Suggested Retail Price. En los cambios introducidos por Jan Kiefer en el 2004, se ve que cambia este valor de Numeric con 2 decimales a decimal por lo que salían muchos decimales. Nosotros agregamos una función en el archivo TPCWPopulate.java para redondear dichos decimales a 2. Para redondear se hace uso de la clase de java BigDecimal y su método Rounding Mode en HALF\_EVEN, que no es más que, valores desde el 5 se redondean hacia arriba y si no, se dejan tal y como están. Este redondeo se aplica también en los decimales de los pedidos tras calcular el descuento si se aplica y en todos aquellos valores que excedan de 2 decimales.
5. Se modifica el servlet TPCWOrderDisplay.java el cual no tenía bien cerrado el “out.print” y no se mostraban los botones de HOME y SEARCH. En la especificación se marca como obligatorio que dichos elementos han de mostrarse en su correspondiente página web.
6. Para los errores de consultas sql se han modificado las queries correspondientes en el archivo sql.properties de TPC-W. Para la búsqueda por título se sustituye la query doTitleSearch:

```
SELECT * FROM item, author
WHERE item.i_a_id = author.a_id AND
```

```
        substring(soundex(item.i_title),0,4)=substring(soundex(?),0,4)
ORDER BY item.i_title limit 50
```

Por:

```
SELECT * FROM item, author
WHERE item.i_a_id = author.a_id AND
item.i_title like ?
ORDER BY item.i_title limit 50
```

Además, se ha modificado el código Java de la clase TPCW-Database.java para que la búsqueda no tenga cuenta los espacios en blanco por delante y detrás de la cadena de búsqueda:

```
statement.setString(1, "%"+search_key+"%");
```

7. Para la búsqueda por autor se sustituye la query doAuthorSearch:

```
SELECT * FROM author, item
WHERE substring(soundex(author.a_lname),0,4)
=substring(soundex(?),0,4) AND
item.i_a_id = author.a_id
ORDER BY item.i_title limit 50
```

Por:

```
SELECT * FROM author, ítem
WHERE (author.a_fname like ? OR author.a_lname
like ? OR concat_ws(' ', author.a_fname,
author.a_lname) like ?) AND item.i_a_id =
author.a_id
ORDER BY author.a_fname,author.a_lname limit 50
```

También se ha modificado el código Java de la clase TPCW-Database.java para que acepte tres parámetros y no tenga en cuenta los espacios en blanco por delante y detrás de la cadena de búsqueda:

```
// Set parameter
statement.setString(1, "%"+search_key+"%");
// Second parameter added to search with author's
name or surname
```

```
statement.setString(2, "%"+search_key+"%");

// Second parameter added to search with author's
name and surname

statement.setString(3, "%"+search_key+"%");
```

De esta forma se puede buscar por nombre, apellido o la combinación de ambos.

La elección del método de búsqueda se realiza mediante un número Random del 1 al 3, que significa: por autor, por título o por categoría. Además, dentro de la búsqueda por autor se genera otro Random para seleccionar si la búsqueda es por nombre, apellido o la combinación de ambos.

Dentro de la búsqueda por título, se implementa de forma que no se hagan búsquedas con palabras de menos de 4 caracteres para evitar búsquedas absurdas y poco reales como buscar una “a”. Además, en las búsquedas, al hacer peticiones mediante URL y estas no son muy amigas de los espacios en blanco, se usa el método Replace de Java para cambiar dichos espacios en blanco por el carácter “+” que equivale al espacio en blanco en la URL de búsqueda.

En este punto, se llega a la conclusión de que, dependiendo del tamaño de los archivos de nombres, apellidos y palabras para la creación de autores y de los títulos de los libros, las búsquedas generadas pueden no ser válidas. Con esto se quiere decir que, si se usa un archivo de 5000 nombres para crear 250 autores, habrá muchos nombres que no saldrán en la función del Random y no serán autores registrados en la base de datos, con lo que luego, también, para las búsquedas obtendremos muy bajo porcentaje de aciertos, ya que pocos registros coincidirán entre tanto valor, situación que dista lejos de la realidad. De esta forma, se ajustan los ficheros de nombres, apellidos y palabras dependiendo del número de ítems, ya que es el factor de escala de TPC-W. A mayor factor de escala, mayor número de registros en los ficheros correspondientes. Todo esto se ha evaluado mediante una función de Java que se ha creado para poder obtener la tasa de aciertos, dependiendo del tamaño de los ficheros usados en cada momento y el factor de escala de la base de datos. Los resultados obtenidos y el tamaño de los ficheros óptimo se muestran a continuación:

- Las tasas de acierto obtenidas están en el rango de un 90%-98% de aciertos en las búsquedas.
- Para un factor de escala de 1.000 se hace uso de un fichero de nombres y apellidos para autores de 100 elementos. El fichero de palabras ha de ser de 1.000 elementos.

- Con un factor de escala de 10.000 se hace uso de un fichero de nombres y apellidos para autores de 1.000 elementos. El fichero de palabras ha de ser de 1.000 elementos.
- Si el factor de escala es de 100.000, se hace uso de un fichero de nombres y apellidos para autores de 10.000 elementos. El fichero de palabras ha de ser de 5.000 elementos.
- Para el resto de factores de escala, se mantienen estos ficheros ya que una vez se tienen muchos elementos en la base de datos, las búsquedas se acercan a aciertos del 98% casi siempre. De esta forma, para bajar un poco esta tasa de aciertos a la hora de buscar por autor, se hace un cálculo de probabilidades que permite elegir si se busca por nombre (48%), por apellido (48%) o por nombre y apellido (6%). Así se obtiene una tasa algo más baja de aciertos planteando una situación algo más realista.

Así, dependiendo del tamaño de la base de datos, aseguramos una tasa de acierto en las búsquedas que se ajustan a búsquedas reales que se pueden efectuar en cualquier e-commerce hoy en día

Para comprobar que las búsquedas se han cambiado bien y se realizan de forma correcta se hizo lo siguiente: cuando las búsquedas no funcionaban devolvían siempre 50 elementos con lo que el peso de la página devuelta rondaba los 12971 bytes. Una vez realizadas las modificaciones, se comprueba que el tamaño de las respuestas varía según el elemento a buscar y el tamaño de los elementos que devuelve. Ahora se puede apreciar como este tamaño varía yendo desde los 2274 bytes para pocos resultados devueltos, pasando por 6453 bytes con unos 25 resultados, para devolver sobre los 12400 bytes en búsquedas que devuelven 50 resultados.

Además, se comprobó navegando desde la página web y haciendo diferentes búsquedas que realmente funcionaban correctamente después de realizar los cambios.

8. También se modifican ciertos archivos de los EB's para adecuar estas soluciones a los navegadores emulados por TPC-W. Se modifica el archivo `rbe.EBWBuReqTrans.java` que es el encargado de generar los nombres, apellidos, emails y demás valores cuando se registra un nuevo usuario. Y también, se modifica el archivo `TPCW_execute_search.java` para que coja las palabras de búsqueda de los ficheros tanto, por nombre o apellidos de autores, como por título de libro como se ha expuesto anteriormente. Esto se ha hecho añadiendo ciertas funciones al archivo `RBE.java` y luego se pueden llamar desde los EB's.

### 3.5. Validación de los cambios efectuados

En este apartado se recogen y analizan los datos obtenidos de varios experimentos en TPC-W, para ver si podemos afirmar que, los cambios introducidos en TPC-W, no afectan al funcionamiento de este benchmark y de esta forma seguir con la adaptación de la base de datos y el entorno a Prestashop.

Se realizarán varios experimentos para cada tipo de usuario (Browsing, Ordering y Shopping) para un número de 10, 20 y 30 usuarios con una duración de 30 minutos de experimento real cada uno. La forma de validar dichos datos se ha realizado ejecutando el mismo experimento en TPC-W sin cambios que en el TPC-W modificado por nosotros, para después ver que los resultados obtenidos son similares en cuanto a peticiones de cada tipo de página web, los WIRT y los WIPS. A continuación, se presentan los resultados de dichos experimentos y las conclusiones que podemos deducir de los valores obtenidos.

Para 10 usuarios tenemos las siguientes gráficas:

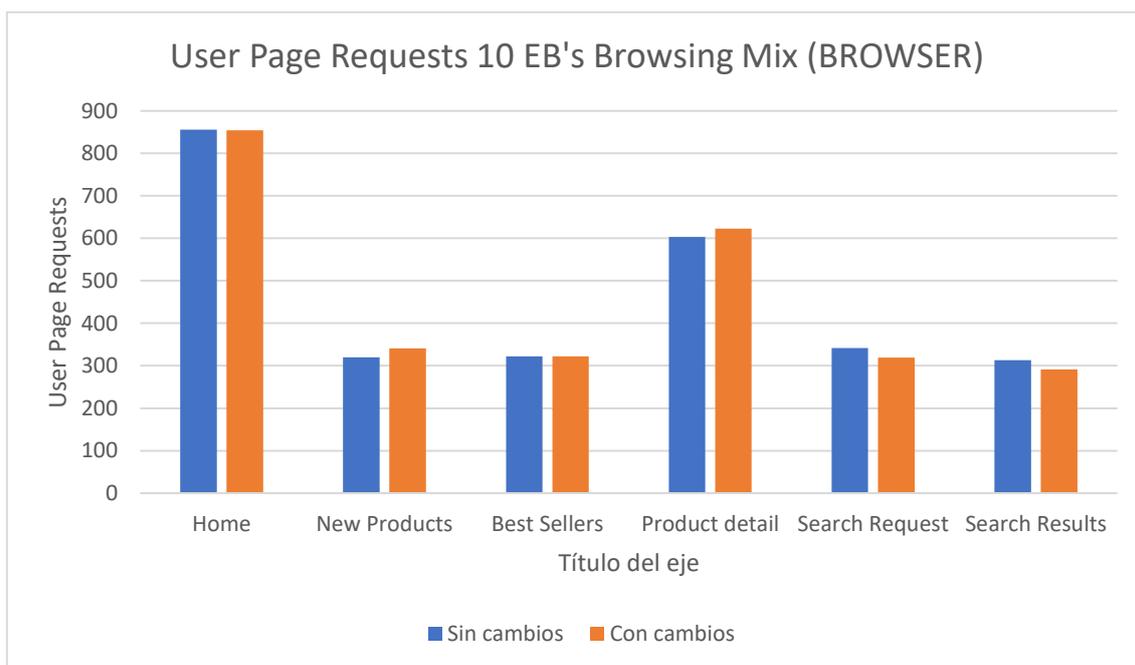


Figura 14: Peticiones web de los usuarios para el browsing mix (BROWSER)

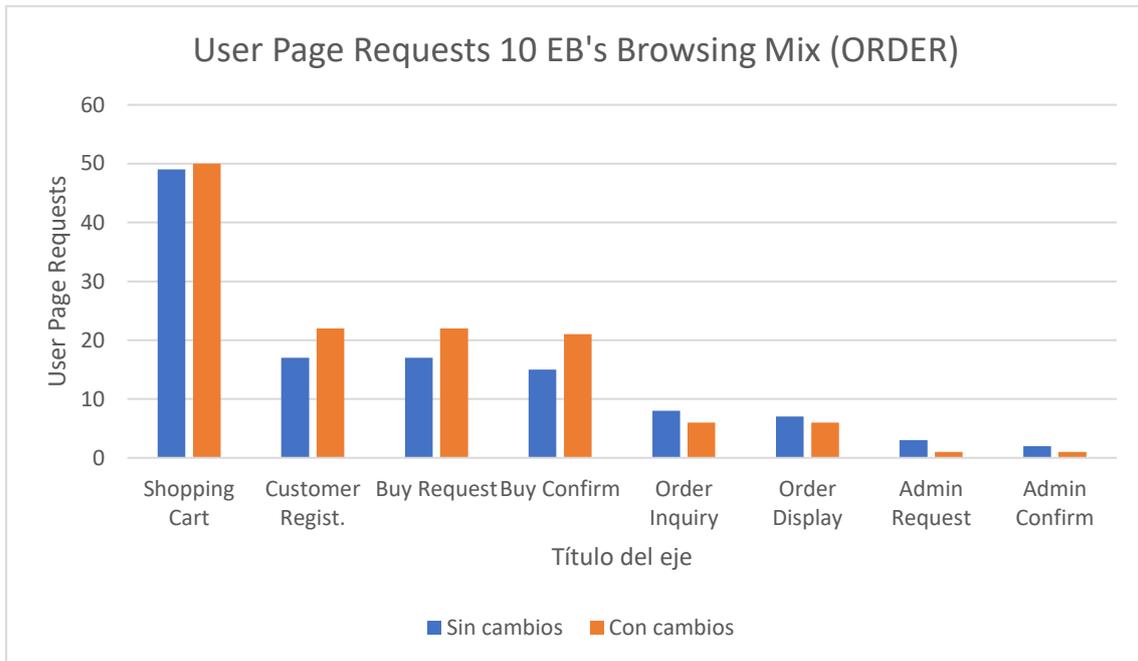


Figura 15: Peticiones web de los usuarios para el browsing mix (ORDER)

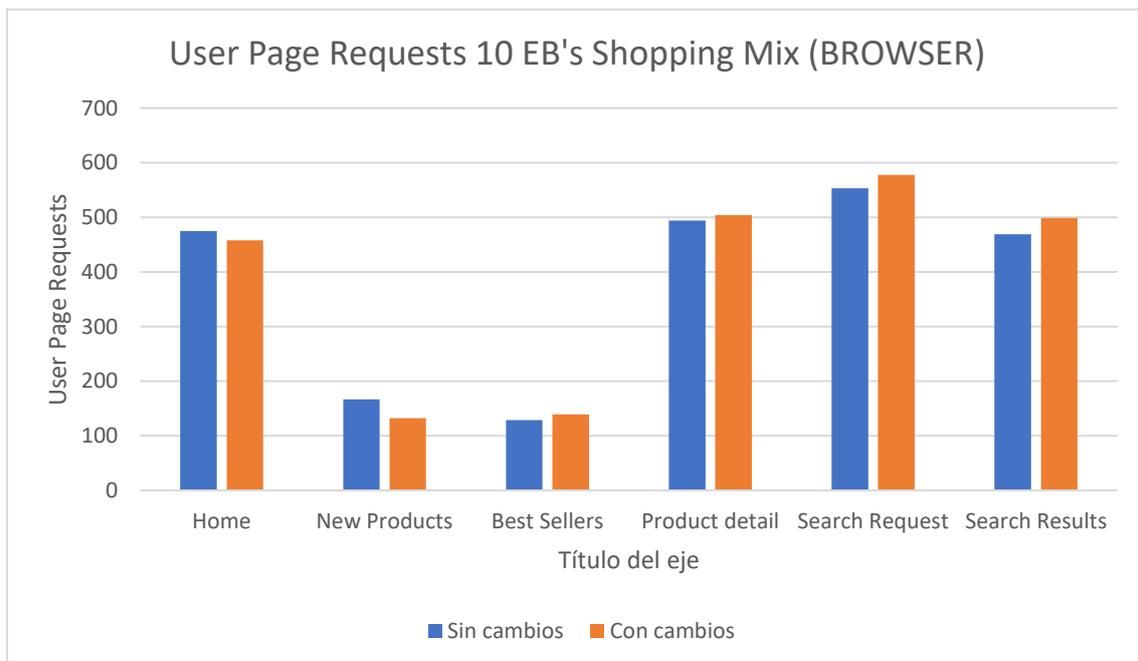


Figura 16: Peticiones web de los usuarios para el shopping mix (BROWSER)

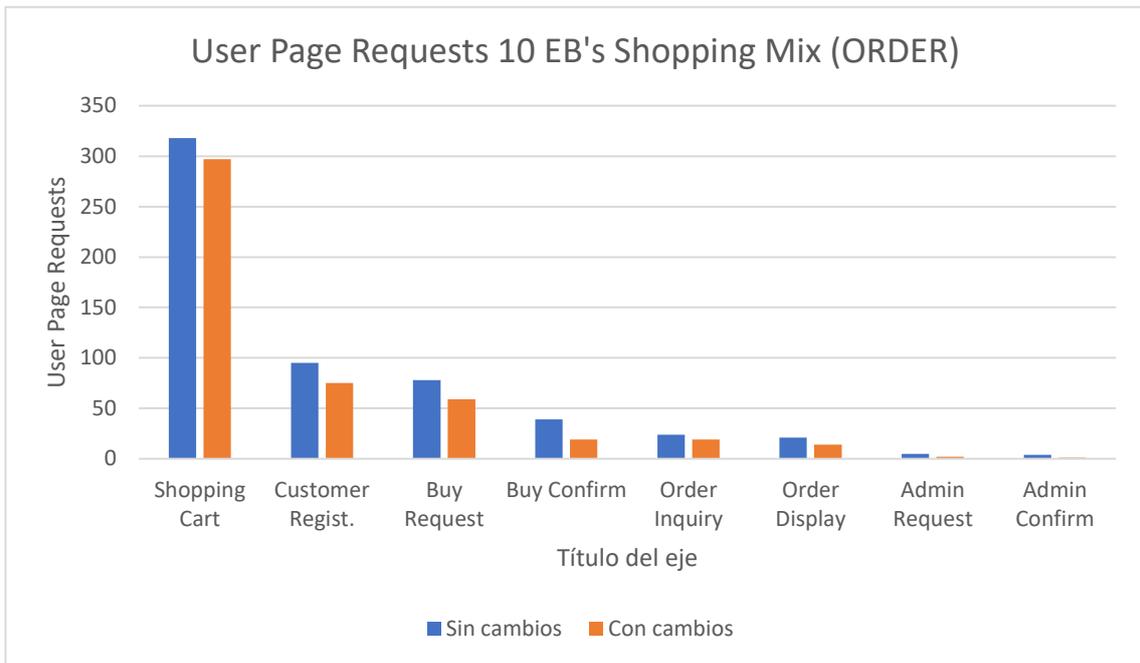


Figura 17: Peticiones web de los usuarios para el shopping mix (ORDER)

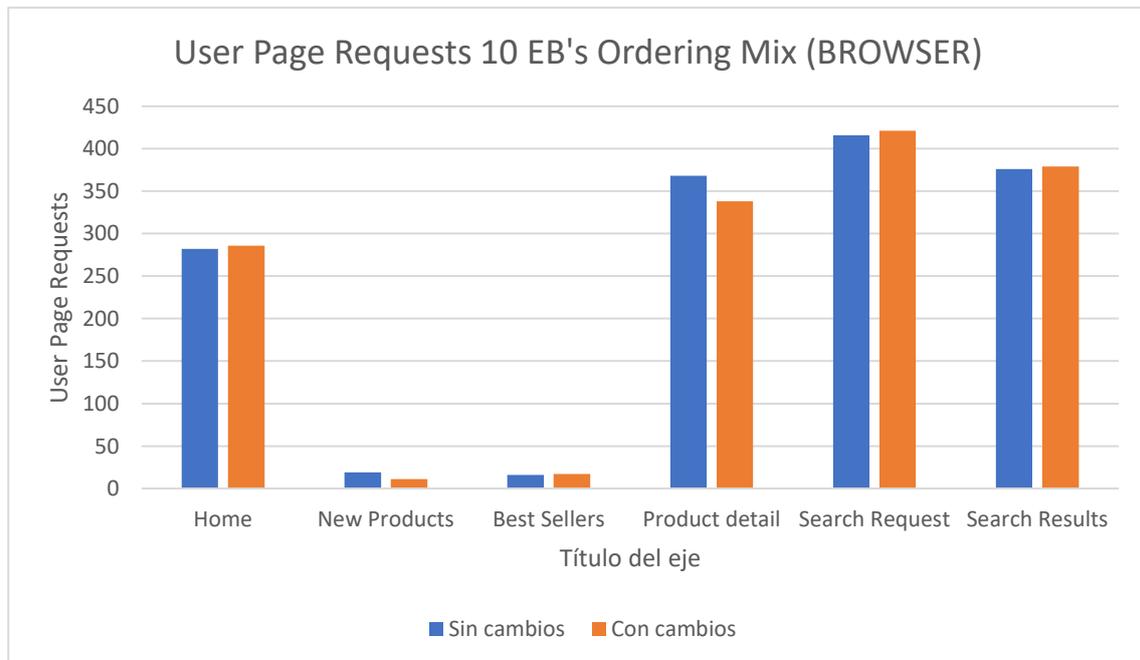


Figura 18: Peticiones web de los usuarios para el ordering mix (BROWSER)

Adaptación de un Benchmark para e-commerce bajo plataformas actuales de ventas como Prestashop haciendo uso del generador de carga GUERNICA

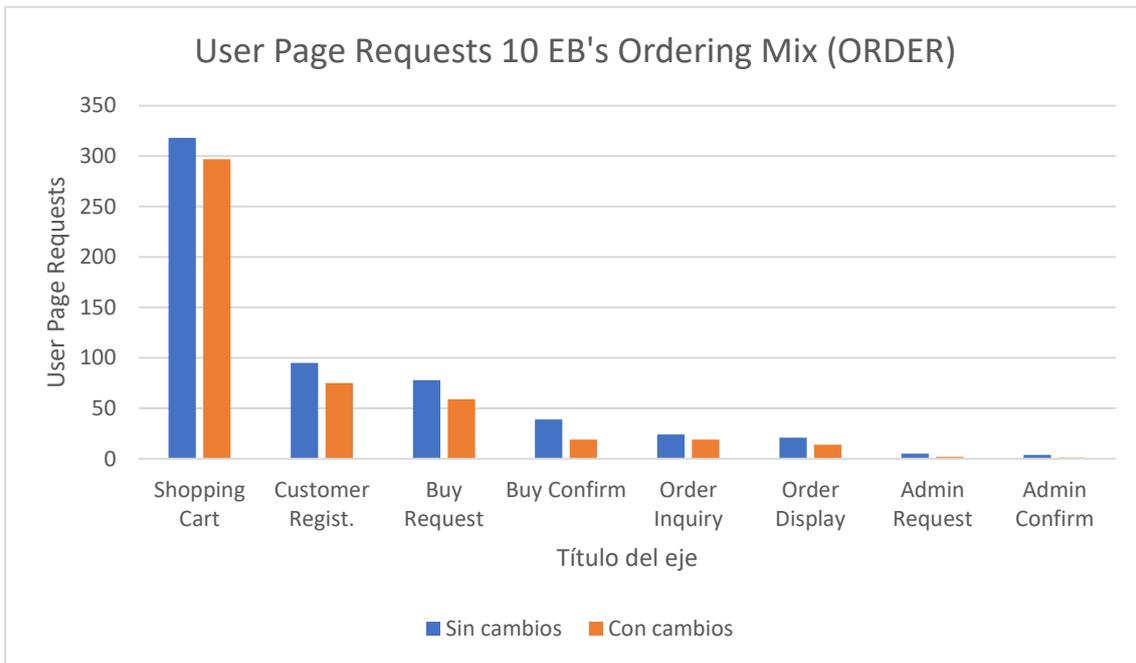


Figura 19: Peticiones web de los usuarios para el ordering mix (ORDER)

Para 20 y 30 EB's se han obtenido gráficas similares a estas.

Además, se han comprobado los WIPS y el Think to Time. A continuación, se muestran ambos valores para 10 EB's de tipo Browsing.

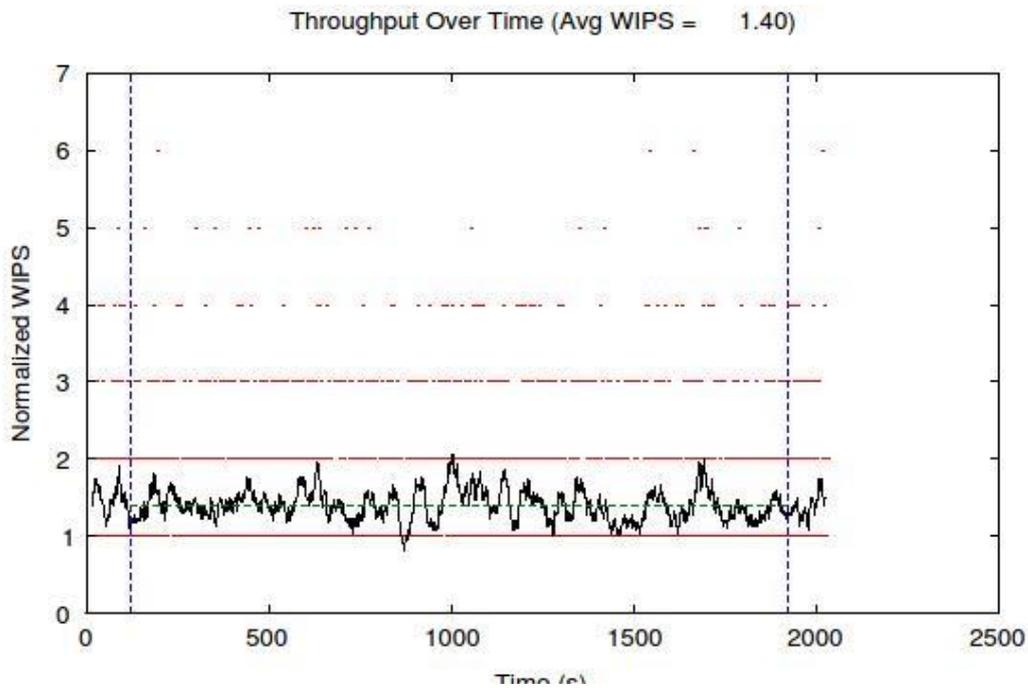


Figura 20: WIPS en TPC-W sin cambios

Adaptación de un Benchmark para e-commerce bajo plataformas actuales de ventas como Prestashop haciendo uso del generador de carga GUERNICA

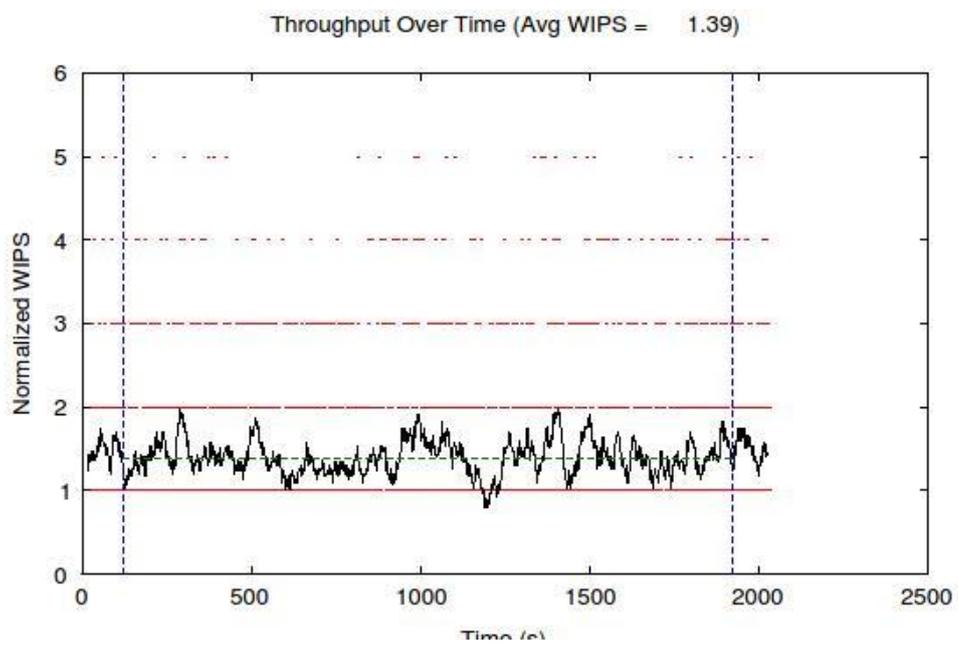


Figura 21: WIPS en TPC-W con cambios

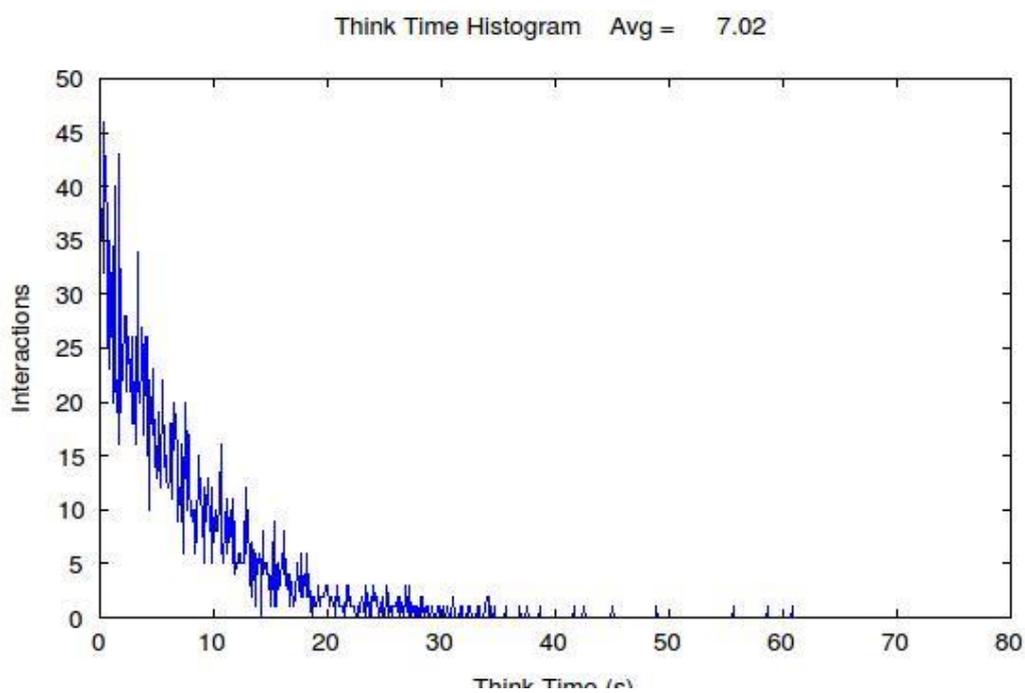


Figura 22: Tiempo de pensar sin cambios en TPC-W

Adaptación de un Benchmark para e-commerce bajo plataformas actuales de ventas como Prestashop haciendo uso del generador de carga GUERNICA

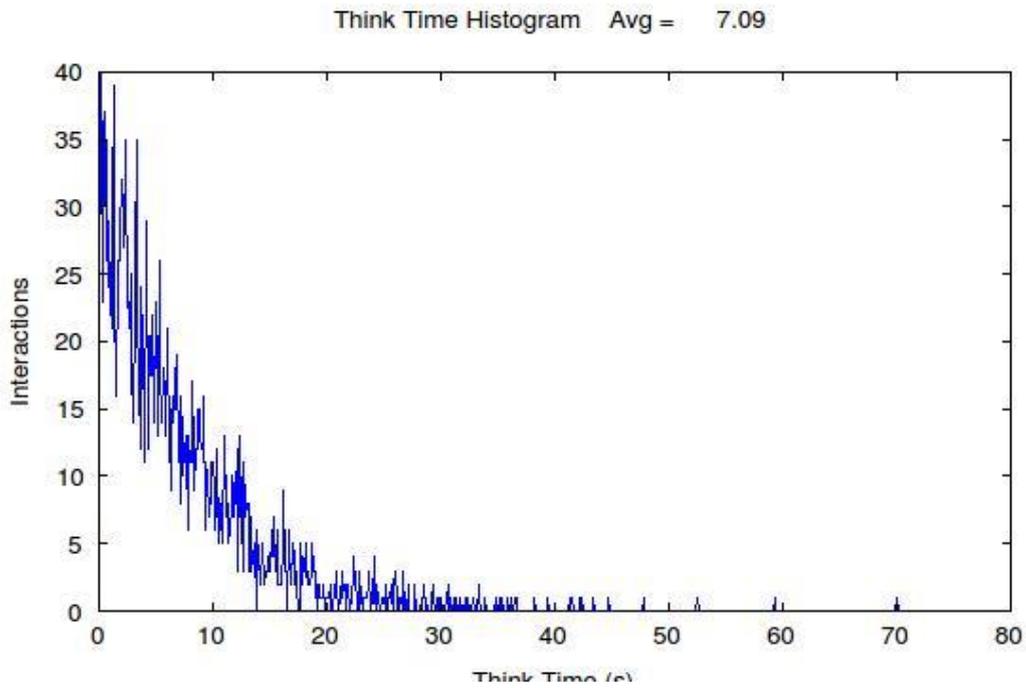


Figura 23: Tiempo de pensar con cambios en TPC-W

Al igual que con las páginas solicitadas por los usuarios, los WIPS y los tiempos de pensar de los usuarios Shopping y Ordering resultan parejos a los expuestos anteriormente. De misma forma, los resultados son similares para 20 y 30 EB's.

Dado los resultados obtenidos podemos decir que realmente no afectan los cambios introducidos a TPC-W, respetándose su especificación y su funcionamiento. De esta forma, obtenemos un entorno más amigable, mucho más fácil a la hora de depurar la base de datos y luego, todo esto, nos servirá para cuando validemos las navegaciones en el CMS, ya que es más fácil reconocer un nombre como "José Montoya" que ver "jksu&"56@er43". Lo mismo pasa con el resto de mejoras introducidas, nos facilitan la depuración de todo el entorno de una forma mucho más fácil.

## 4 Prestashop

En este capítulo se presenta la plataforma de Prestashop. Se hablará un poco de este CMS, su instalación, la migración de la base de datos de TPC-W a Prestashop y las modificaciones necesarias para un correcto funcionamiento. Por último, verificaremos a través del panel de control que todo está correcto.

### 4.1. Descripción

Prestashop es un CMS muy popular actualmente utilizado para la creación de páginas web de tipo e-commerce. Tiene ya unos cuantos años y unas cuantas versiones a sus espaldas (1.4, 1.5, 1.6 y actualmente la 1.7). Para este proyecto se hace uso de la versión 1.16.15. Esta plataforma hace uso de MySQL, PHP y el servidor Apache altamente utilizado en la actualidad como se puede apreciar en la siguiente imagen:

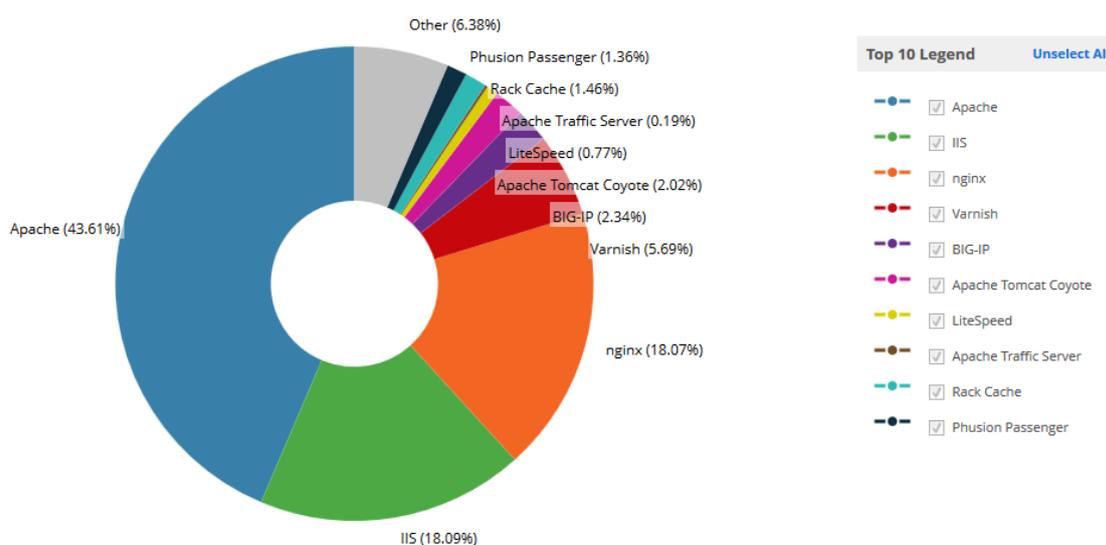


Figura 24: Presencia actual de las principales plataformas de servidores  
Fuente: [5] 2017

Es uno de los CMS más utilizados a nivel europeo y sobre todo en España. Permite crear páginas web de e-commerce con un aspecto muy profesional y siendo de código libre. Además, como se dijo en el capítulo 2, goza de una amplia comunidad de usuarios, desarrolladores y entusiastas de este CMS. Hoy en día existen miles de páginas creadas bajo este CMS y en más de una de ellas, algún conocido o nosotros mismos, habremos efectuado alguna compra en los últimos años.

Además, este CMS es bastante fácil de gestionar a través de su panel de control o backend el cual nos permite realizar múltiples acciones como administrador (añadir artículos, ver pedidos, crear usuarios, añadir categorías, etc.) y también ver ciertos parámetros de nuestro sitio (visitantes online, carritos de compra abandonados, usuarios nuevos registrados, etc.). Se trata de un entorno muy amigable que se gestiona como si de una página web como el correo de Google se tratara. Ofrece miles de opciones y configuraciones posibles y uno de los aspectos más importantes es la posibilidad de añadir mediante archivos csv ciertos parámetros de forma masiva en su base de datos. Entre estos se encuentra la tabla de clientes, la tabla de direcciones, la tabla de artículos, la tabla de fabricantes o proveedores, las categorías y alguno más. De esta forma se realizará la adaptación de la base de datos de TPC-W a la base de datos de Prestashop.

Para este proyecto no se hace uso de ningún modulo externo ni plantilla de tema para el diseño, sino que se usa el que trae la instalación por defecto al igual que los módulos básicos de Prestashop.

## **4.2. Instalación**

Para la instalación de este CMS se clona la máquina virtual que tenemos con TPC-W, de esta forma se mantienen las características físicas de la máquina al igual que las versiones de MySQL y Java, siendo estas las mismas en ambos entornos. Los cambios que se realizan son la desinstalación de Apache Tomcat, la eliminación de la base de datos usada por TPC-W y la instalación de Apache la cual es necesaria para el funcionamiento de Prestashop.

Lo primero que se necesita es instalar Apache y PHP, además de unos cuantos módulos adicionales de PHP. Una vez instalados lo que tenemos que hacer es descargar la versión de Prestashop y descomprimirla en la raíz del servidor Apache suele ser `www/html`.

Una vez alojados los archivos en la raíz de nuestro servidor solo tenemos que indicar la ruta donde estén en cualquier explorador web y Prestashop nos permitirá efectuar una instalación de forma guiada mediante pasos. En caso de que falte algún componente Prestashop nos dirá cual está fallando y es necesario instalar. En medio del proceso de instalación se ha de crear una base de datos, haciendo uso de un usuario que no sea root para realizar las acciones sobre esta base de datos por parte de Prestashop. Estos datos se pasan a la instalación del CMS que configura todo, saltando algún aviso en caso de error y sino, indicando que todo está correcto.

Después de todo el proceso Prestashop nos pedirá que eliminemos los archivos de instalación para evitar sobreinstalaciones por parte de terceros. De esta forma, una vez eliminados dichos archivos ya tendremos acceso al backend de Prestashop introduciendo nuestro mail y password de administrador. Además, ya podemos ver el frontend en el explorador y ver cómo es la página web creada.

Como se dijo en el capítulo 2 la instalación de Prestashop es fácil e intuitiva, siendo guiada prácticamente. En caso de querer ver más detalles de la instalación se recomienda consultar la documentación oficial [16].

Adaptación de un Benchmark para e-commerce bajo plataformas actuales de ventas como Prestashop haciendo uso del generador de carga GUERNICA

El backend de Prestashop tiene un aspecto como se puede ver en la figura 25:

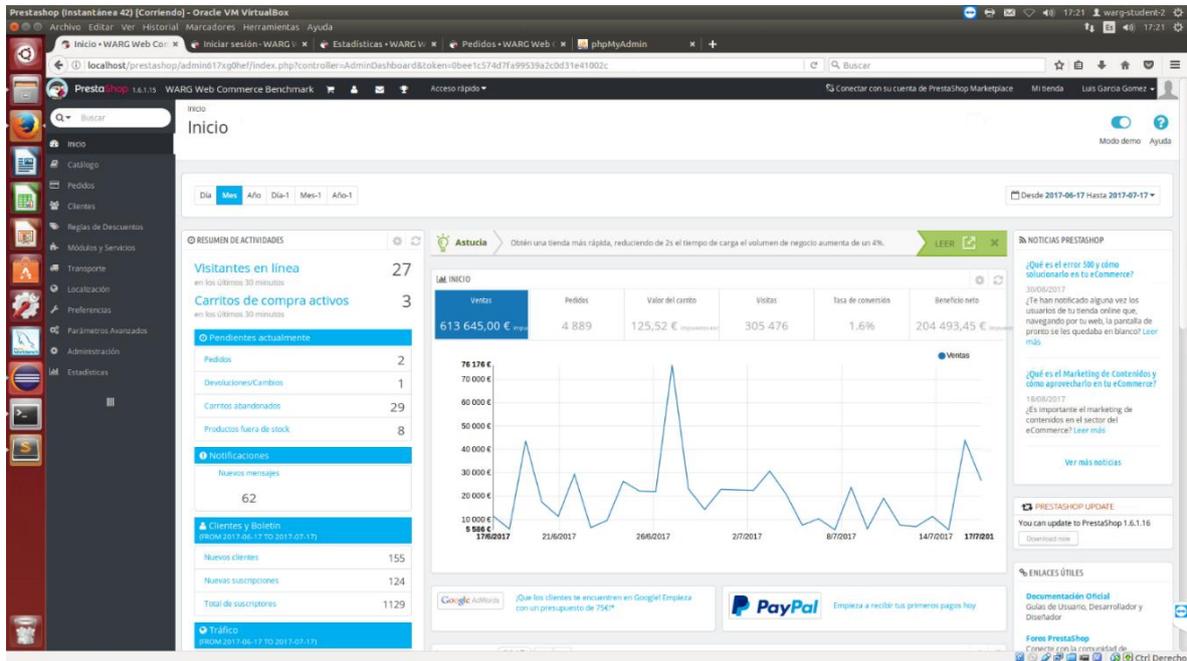


Figura 25: Panel de control de Prestashop

Y la página web que muestra o frontend lo podemos ver en la figura 26:

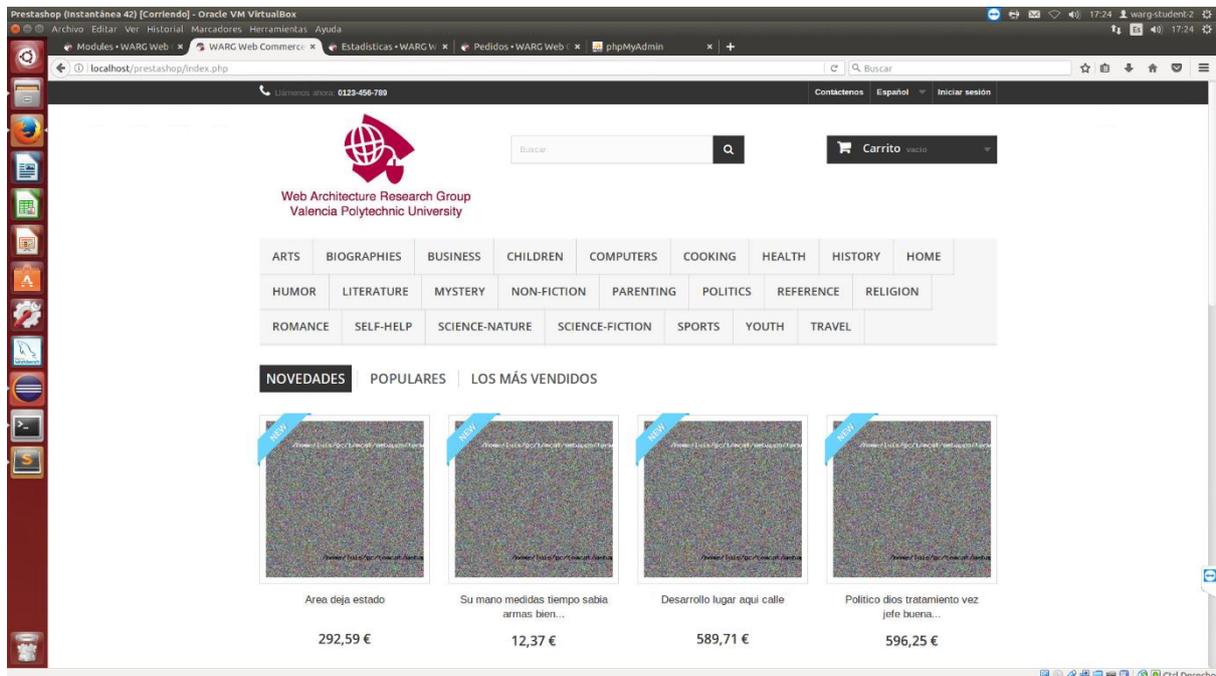


Figura 26: Frontend de Prestashop

### 4.3. Popularización de la base de datos

Se ha de puntualizar que la implementación de la base de datos de Prestashop es mucho más compleja que la de TPC-W, la cual consta solo de 10 tablas. Un vistazo rápido de la base de datos de este CMS, donde se puede apreciar su nivel de complejidad, es la siguiente imagen:

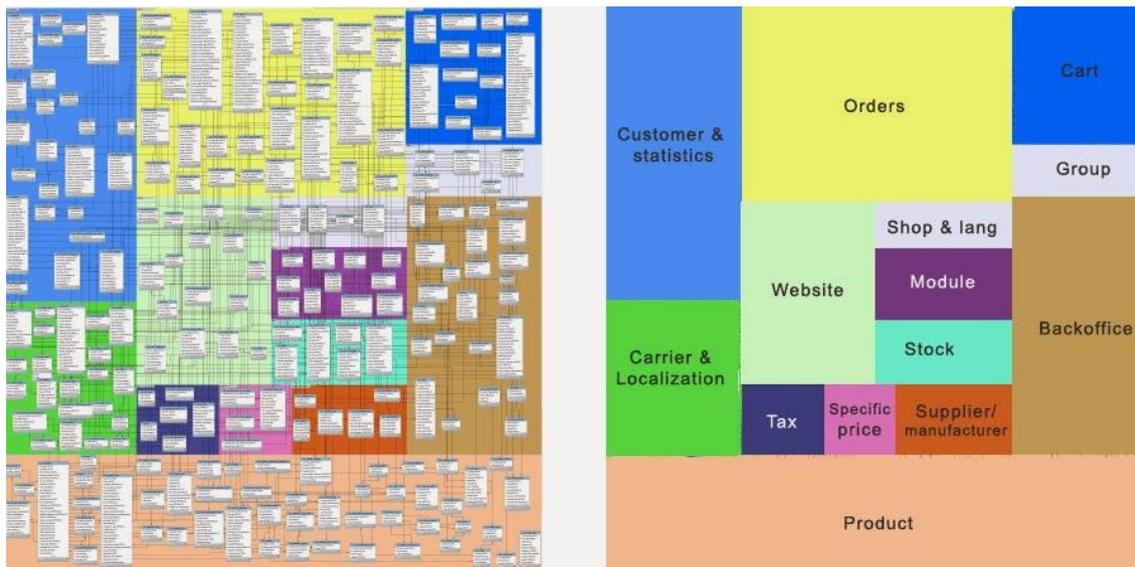


Figura 27: Base de datos de Prestashop

Por este motivo, se ha adaptado la base de datos a la de Prestashop de forma consecuente. Las tablas de `shopping_cart` y `shopping_cart_line` no las rellena TPC-W, se rellenan en las navegaciones de dicho benchmark. Ambas tablas existen en Prestashop y, además, no hay tablas para guardar números de tarjetas de crédito, la de direcciones funciona de otra forma y la de pedidos no se puede rellena mediante un archivo csv.

Como se ha dicho anteriormente se hará uso de archivos csv para rellena la base de datos de forma masiva. Para ello, se cogen archivos de ejemplo que trae Prestashop y se ven los campos necesarios con aquellos campos que nos obligan a rellena. Lo primero que nos fijamos al hacer una pequeña navegación de prueba por la página es que, los password en Prestashop han de ser de 5 letras con lo que necesitamos cambiar los 99 primeros clientes por valores que cumplan dicho requisito. Esto se hace de forma fácil al adaptar la tabla de clientes de TPC-W al formato del csv. Lo más relevante que observamos es que el valor 3 en el group del cliente se corresponde con usuario registrado, (1 es visitante, por ejemplo), por eso al popularizar la tabla de clientes hemos de poner este valor a 3 en todas las filas del csv ya que son clientes registrados. De esta forma, solo nos queda ir al panel de control de Prestashop e importar dicho archivo. Una vez elegido el archivo que queremos subir le damos a importar y se rellena la tabla de clientes. En caso de algún error en la importación Prestashop nos dará

información de que registros y que campos no se han importado de forma correcta.

Si nos fijamos en el apartado de clientes en el backend, podremos ver el número de clientes que hay, en este caso tenemos 28801 y una lista con los nombres y algunos datos más de cada uno de estos clientes. La importación se ha realizado con éxito en este punto, como se observa en la figura 28:

ID	Tratamiento	Nombre	Apellidos	Dirección de correo electrónico	Ventas	Activado	Boletín de noticias	Opt-in
28801	--	Direcciones	Fantasma	fantasma@mioperador.com	--	✗	✗	✗
28800	--	Guillen	Pellon	ALININBABA@miempresa.com	--	✓	✗	✗
28799	--	Basilio	Bustelo	ALINULNGNG@inbox.com	--	✓	✗	✗
28798	--	Jose	Santisteba	ALINULNGIN@yahoo.com	--	✓	✗	✗
28797	--	Bruno	Leys	ALINULNGUL@movistar.com	--	✓	✗	✗
28796	--	Victor	Pons	ALINULNGAT@zoho.com	--	✓	✗	✗
28795	--	Demetria	Amirola	ALINULNGSE@s-mail.com	--	✓	✗	✗
28794	--	Filiberto	Gos	ALINULNGRE@terra.com	--	✓	✗	✗

Figura 28: Clientes después de importar el archivo csv

Para el archivo de direcciones y conseguir los valores correctos, hace falta realizar una concatenación (JOIN) de varias tablas de TPC-W, de forma que con las tablas de clientes, direcciones y países se pueden adaptar a las direcciones de Prestashop. De esta forma, se crea el archivo csv para las direcciones. En este punto, si se intenta importar, lanzará errores de que varios registros no se han registrado de forma correcta. Esto es debido a que el nombre de los países que usa TPC-W están en inglés y cuando se importan, hay alguno de ellos que no se traduce de forma correcta y da error, al igual que uno de ellos no está registrado en Prestashop y también da error. Estos países son:

- Farkland Islan → Islas Malvinas
- Bangla Desh → Bangladesh
- Trinidad → Trinidad y Tobago
- Faroe Island → Islas Feroe
- Russia → Rusia, Federacion de
- Luxembourg → Luxemburgo
- Kazakhstan → Kazajistan
- Eastern Caribbean → No existe y se cambia por Ucrania.

Una vez se han realizado las modificaciones, la importación de la tabla de direcciones se hace de forma correcta, aunque se observa que solo hay 28800 registros. Esto se debe a que cuando TPC-W populariza la base de datos crea direcciones aleatorias fijándose solamente en el factor de escalado sin tener en cuenta el número de clientes registrados. De esta manera, se crean direcciones que no tienen ningún cliente asociado y se usan las mismas direcciones para diferentes clientes, cosa que no tiene mucha lógica en un entorno real. Efectivamente, en Prestashop esto no es así, siendo necesaria que una dirección se asocie a un cliente como mínimo.

Por lo anterior, se ha optado por la solución de crear un usuario el cual no estará activado y será el dueño de las 28.800 direcciones restantes. Se hace uso de la concatenación anteriormente usada pero negada y de esta forma se obtienen todas las direcciones que no tienen ningún cliente asociado.

De esta forma, se crea otro archivo csv que contiene estas direcciones y las asocia al cliente no activado creado anteriormente. Se importa este archivo como se hizo con el anterior de direcciones, sin marcar la opción de vaciar la tabla porque si no se borran las direcciones anteriores. Se observa como Prestashop no arroja ningún error y en el apartado de clientes-direcciones vemos como tenemos las 57.600 direcciones en nuestra base de datos como muestra la siguiente imagen:

ID	Nombre	Apellidos	Dirección	Código postal/Zip	Ciudad	País
86037	Direcciones	Fantasma	8F5QEAI5fVxcCkm02Ottv96Aj	HLddOtrnCQs	Flw8fRm	España
86036	Direcciones	Fantasma	TIR24TRxdlHgHPKkajjB	U3wGval	WRNijzYmojBroW5Vyul9OfXPuNi	España
86035	Direcciones	Fantasma	lZchru7ReblAjctZQgeVRbnlHtp	sDIZ6Wh	456YzP8eDEinUp	España
86034	Direcciones	Fantasma	4v4pLT644HrNwOINu0a9QBraIGp	6hPUinUEoY9	tRsnZL6CFT	España
86033	Direcciones	Fantasma	AxFc51V6ApZp84Kn0D6OYktCro	PSq9cy	2xBBBWGKjxnhjBAX49DwxU0	España
86032	Direcciones	Fantasma	ObxZg0MXE0on01gYM9	fRvLY	oMNNorWz1j8obL	España
86031	Direcciones	Fantasma	uYvKp6LxCBhhWp4j5f3mYaTlu3PPCvxObfUA	hBLVCS	Bv5c	España
86030	Direcciones	Fantasma	GBTN8InLa93tCxcg0dQEHSER3M7e6L	sRzZB	JcsW e Hn4LoVA 7	España
86029	Direcciones	Fantasma	nEUar2PMto1Pb7ZCbgfjdhVM6QFXVL	6wjRW8fH	P1NPHCLIU2uQ	España

Figura 29: Direcciones después de importar su archivo csv

En este punto, se tienen popularizadas las tablas de clientes y direcciones sin ningún problema con el mismo número de registros que TPC-W.

Para los autores, se hace uso de la tabla de fabricantes de Prestashop, y se utiliza el atributo m\_name + la fecha de nacimiento que se genera en TPC-W para rellenar la descripción corta del autor. Una vez creado el csv correspondiente se importa con éxito a Prestashop, figura 30.

ID	Logotipo	Nombre	Direcciones	Productos
--	<input type="checkbox"/>	<input type="text"/>	--	--
<input type="checkbox"/> 250		Cesar Enrique Rodrigues	--	6
<input type="checkbox"/> 249		Alberto Vicente Romero	--	3
<input type="checkbox"/> 248		Carlos Mariano Hu	--	2
<input type="checkbox"/> 247		Angela Cristina Zhang	--	3
<input type="checkbox"/> 246		Angelica Liu	--	6
<input type="checkbox"/> 245		Safwan Da Silva	--	4
<input type="checkbox"/> 244		Ivan Antonio Rodriguez	--	4
<input type="checkbox"/> 243		Laura Victoria Molina	--	7

Figura 30: Autores después de importar su archivo csv

Además, se han de crear las categorías correspondientes. En TPC-W se tienen 24 categorías, y estas se pueden importar mediante un csv. Una vez importadas las categorías, estas se han de activar para que se muestren en el menú horizontal de la página web. Esto se realiza yendo al apartado de módulos del backend, buscar “menú horizontal” y configurar para que se muestren las categorías que hemos añadido.

En la importación de productos se han de hacer algunos cambios antes de proceder. En TPC-W el estado del artículo tiene 5 opciones (hardback, paperback, used, audio y limited-edition), en cambio en Prestashop solo dispone de 3 (new, used, refurbished). Por lo que se ha cambiado de la siguiente manera:

- HARDBACK → NEW
- PAPERBACK → USED
- USED → USED
- AUDIO → REFURBISHED
- LIMITED\_EDITION → NEW

El impuesto en TPC-W es del 8,25% en Prestashop tiene el valor del 21%. También se añaden características al producto al igual que se tienen en TPC-W, se muestran:

- Publicado por.
- La fecha de publicación.
- El ISBN.
- Las páginas.

Adaptación de un Benchmark para e-commerce bajo plataformas actuales de ventas como Prestashop haciendo uso del generador de carga GUERNICA

- Y las dimensiones en pulgadas en formato (22x34x33).

Para la descripción corta se ha acotado la descripción larga en 100 caracteres desde el inicio. Luego se procede a su importación y una vez Prestashop nos diga que todo está correcto, se puede observar en el apartado fabricantes que hay 1000 entradas, figura 31.



ID	Imagen	Nombre	Referencia	Categoría	Precio base	Precio final
1		Area diferencia fue claro podemos nosotros quiero		BIOGRAPHIES	463,40 €	560,71 €
2		Haya republica grupo hemos don		BIOGRAPHIES	464,10 €	561,56 €
3		Encuentra internacional padres juicio visita trata		SCIENCE-NATURE	522,42 €	632,13 €
4		Productos		SCIENCE-FICTION	645,37 €	780,90 €
5		Tiene resulta trabajadores nuestro dentro realidad perdido		COOKING	7,04 €	8,52 €

Figura 31: Producto después de importar su archivo csv

La carpeta de imágenes generadas en TPC-W para los artículos se ha de copiar al directorio de Prestashop y aparte, cambiar el propietario de esta carpeta de imágenes a www-data, (chown -R www-data:www-data) ya que es el usuario con permisos dentro de Prestashop. Hay que recordar que, una vez importados los artículos, hemos de reconstruir los índices para que los nuevos productos se indexen en las búsquedas, esto se hace a través del backend desde Preferencias→Buscar→Reconstruir el índice completo.

Una vez acabadas todas las importaciones, se ha conseguido obtener el mismo número de registros para clientes, autores, direcciones y artículos que TPC-W en la base de datos de Prestashop. La tabla de países es una tabla por defecto de Prestashop con un mayor número de registros que la de TPC-W y la de pedidos, esta será rellenada más tarde mediante navegaciones desde GUERNICA. En este punto, se puede decir que tenemos el mismo punto de partida, a falta de los pedidos, en ambas plataformas.

Una de las cosas a destacar es que, gracias a las modificaciones iniciales realizadas sobre la popularización de TPC-W, se ha conseguido adaptar casi sin problemas la base de datos a Prestashop, evitando muchas violaciones de restricción que impone el CMS y se han obtenido los csv correspondientes con pocas modificaciones adicionales y sin muchas complicaciones.

Para terminar este apartado, se muestra el aspecto que tiene un producto en la página web de Prestashop con la siguiente imagen:

DATA SHEET	
Publicado por	NIKs08IOchJ3I2eTmZYHRm7KYnmLbEhbmhzC
Fecha publicación	1962-10-23
ISBN	Hwx3V0OMoBnx2
Páginas	3496
Dimensiones (en pulgadas)	4.09x7.24x37.09

Figura 32: Aspecto de un producto en la página web de Prestashop

## 4.4. Cambios necesarios

Para que funcione de forma correcta Prestashop, es necesario realizar las siguientes modificaciones desde el panel de control:

Cambiar la cantidad de artículos que se muestran en la página por defecto. El valor por defecto es 12 y se ha aumentado a 24 desde Preferencias → Productos → Paginación. Este parámetro se puede poner al gusto y se ha puesto este valor por mostrar un mayor número de productos, como puede producirse en TPC-W.

También se ha eliminado el DNI obligatorio para las compras desde Localización → Países → España, sino se hace necesaria la generación de números aleatorios de DNI para las compras complicando la adaptación. En este mismo sitio se ha de modificar el código ZIP, que por defecto esta como NNN (significa solo números) y se ha de dejar en blanco para que acepte cualquier formato como código ZIP, y de esta forma que sean válidos los códigos ZIP que se generan en TPC-W.

Otro de los cambios que se han realizado ha sido, mediante una consulta UPDATE, poner todos los países de los clientes como España. Esto se hace porque en Prestashop, se han de habilitar las compras en diferentes países y solo se puede

realizar de forma manual. Además, dentro de cada país, Prestashop muestra un desplegable con las ciudades correspondientes al país elegido. Luego a la hora de crear usuarios, este CMS relaciona los países con las ciudades a través de códigos, por ejemplo, para España es el 6 y usa desde el 313 hasta el 365 para provincias, los cuales corresponden a las diferentes ciudades de España. Si se utilizan más países, se han de averiguar dichos valores para cada país y esto es muy costoso para generar ciudades y países en el registro de usuario. Por eso se ha optado por la solución más sencilla, acotando estos valores y trabajando sobre España, cosa que no implica que la navegación deje de ser realista, lo único que todas las direcciones de los clientes corresponderán a alguna ciudad de España.

De igual forma, se han cambiado todas las contraseñas de los clientes a un único valor siendo este “12345”. Esto facilitará las cosas cuando se loguea un usuario mediante navegación, no siendo necesario el mantener todas las claves en algún registro ya que Prestashop encripta los passwords en su base de datos.

## **4.5. Resultado final**

Una vez finalizados los pasos anteriores, se obtiene un entorno listo para empezar a crear las diferentes navegaciones de los clientes y hacer uso de GUERNICA.

De esta forma, se parte de un estado de base de datos en Prestashop similar al punto de partida de TPC-W al popularizar su base de datos. La única tabla que queda por rellenarse es la de pedidos que, como se ha comentado antes, se rellenará mediante GUERNICA haciendo una navegación exclusiva de compra para rellenarla lo más rápido posible.

Por el resto, se mantienen las especificaciones de TPC-W a la hora de popularizar su base de datos a Prestashop. En caso de querer escalar Prestashop lo que es necesario es popularizar TPC-W con el número de artículos o ítems deseados y una vez se ha realizado el paso anterior, lo único que quedaría sería generar los ficheros csv e importarlos en Prestashop. Hay que puntualizar que a mayor número de artículos o ítems más se tardará en llevar a cabo este proceso de popularización.

A partir de todo lo anterior, se efectúan las modificaciones correspondientes en TPCWPopulate.java para no tener que realizar las adaptaciones vistas en el apartado 4.3 cada vez que se genera la base de datos de TPC-W.

En este punto, se tiene la misma página de e-commerce bajo Apache Tomcat y TPC-W que la que tenemos bajo el entorno del CMS Prestashop. Esto quiere decir que las navegaciones, las compras, el registro de usuarios, los nombres de autores, los títulos de los libros, el número de elementos en las tablas de las bases de datos, etc., son idénticos. En este momento, se podría realizar la misma compra con el mismo artículo por parte del mismo cliente en ambas plataformas.

## 5 GUERNICA

---

A continuación, se presenta el generador de carga GUERNICA, como se crean las navegaciones y los métodos de código Java para realizar las diferentes acciones de los usuarios. Además, se muestra cómo se han validado las navegaciones a través del panel de control de Prestashop.

### 5.1. Descripción

GUERNICA es un generador de carga desarrollado por el grupo de investigación, el cual está basado en TPC-W y que, además, se puede decir que este generador ha sido ya validado bajo este estándar como se observa en el trabajo presentado por Raúl Peña en [6]. Hace uso de una librería independiente de Java llamada TGI para hacer la integración de TPC-W con GUERNICA. Todo este trabajo puede consultarse como se ha dicho anteriormente en [6], además de ver que este se validó con TPC-W obteniendo resultados con éxito y de tal forma podemos decir que: es un generador de carga dinámico contrastado y validado frente al estándar TPC-W. Se cita del trabajo de Raúl Peña Ortiz [6]:

*“Dweb y GUERNICA pueden generar cargas de trabajo tradicionales precisas para los estudios de rendimiento web basados en TPC-W. Además, debido a sus diseños, GUERNICA puede utilizarse para generar cargas de trabajo web con el comportamiento dinámico de los usuarios.”*

### 5.2. Navegaciones

La instalación de GUERNICA y TGI se realiza sobre el entorno Eclipse, ya que la parte de TGI, que se encarga de las acciones, se desarrolla en Java y, además, se han de hacer las correspondientes modificaciones para las navegaciones de los usuarios. De esta forma también, se puede realizar debug desde el entorno de desarrollo eclipse al lanzar carga con el generador, muy útil para la depuración de fallos.

Las navegaciones se generan conforme la especificación de GUERNICA como se propone en [6].

Se generan los archivos de navegación para los usuarios de Browsing, Ordering y Shopping. Estos archivos son de tipo XML (Extended Markup Language) y en estos archivos, se han de poner los porcentajes que tiene cada transición entre estados o páginas. Estos valores son distintos para cada tipo de usuario y se han obtenido a partir de la especificación de TPC-W, donde se tienen las siguientes tablas con los valores correspondientes:

Adaptación de un Benchmark para e-commerce bajo plataformas actuales de ventas como Prestashop haciendo uso del generador de carga GUERNICA

To this Web Interaction → From this Response Page	Admin Confirm	Admin Request	Best Sellers	Buy Confirm	Buy Request	Customer Regist.	Home	New Products	Order Display	Order Inquiry	Product Detail	Search Request	Search Results	Shopping Cart
Admin Confirm							9952					9999		
Admin Request	8999						9999							
Best Sellers							167				472	9927		9999
Buy Confirm							84					9999		
Buy Request				4614			6546							9999
Customer Regist.					8666		8760					9999		
Home			3124					6249		6718		7026		9999
New Products							156				9735	9784		9999
Order Display							69					9999		
Order Inquiry							72		8872			9999		
Product Detail		58					832				1288	8603		9999
Search Request							635						9135	9999
Search Results							2657				9294	9304		9999
Shopping Cart						2585	9552							9999

Tabla 2: Umbrales para el intervalo de compras

Fuente: [15]

To this Web Interaction → From this Response Page	Admin Confirm	Admin Request	Best Sellers	Buy Confirm	Buy Request	Customer Regist.	Home	New Products	Order Display	Order Inquiry	Product Detail	Search Request	Search Results	Shopping Cart
Admin Confirm							9877					9999		
Admin Request	8999						9999							
Best Sellers							4607				5259	9942		9999
Buy Confirm							342					9999		
Buy Request				9199			9595							9999
Customer Regist.					9145		9619					9999		

Adaptación de un Benchmark para e-commerce bajo plataformas actuales de ventas como Prestashop haciendo uso del generador de carga GUERNICA

Home			3792					7585		7688		9559		9998
New Products							299				9867	9941		9999
Order Display							802					9999		
Order Inquiry							523		8856			9999		
Product Detail		47					8346				9749	9890		9999
Search Request							788						9955	9999
Search Results							3674				9868	9942		9999
Shopping Cart						4099	8883							9999

Tabla 3: Umbrales para el intervalo de compras

Fuente: [15]

To this Web Interaction → From this Response Page	Admin Confirm	Admin Request	Best Sellers	Buy Confirm	Buy Request	Customer Regist.	Home	New Products	Order Display	Order Inquiry	Product Detail	Search Request	Search Results	Shopping Cart
Admin Confirm							8348					9999		
Admin Request	8999						9999							
Best Sellers							1				333	9998		9999
Buy Confirm							2					9999		
Buy Request					7999		9453							9999
Customer Regist.						9899	9901					9999		
Home			499					999		1269		1295		9999
New Products							504				9942	9976		9999
Order Display							9939					9999		
Order Inquiry							1168		9968			9999		
Product Detail		99					3750				5621	6341		9999
Search Request							815						9815	9999
Search Results							486				7817	9998		9999
Shopping Cart						9499	9918							9999

Tabla 4: Umbrales para el intervalo de compras

Fuente: [15]

Esta tabla se interpreta de la siguiente forma:

Se genera un número aleatorio entre 0 y 9999, una vez obtenido este número si no está en la tabla se elige el siguiente número mayor que si está en la tabla. El que no haya ningún valor en la celda significa que no hay transición posible desde la página actual hacia esa página. Por ejemplo, si estamos en el Shopping Interval, tenemos el número 6371 y estamos en Home, el número más cercano y mayor es el 6718, esto significa que transitaremos a la página de Order Inquiry. Como se puede observar hay celdas que no contienen ningún valor con lo que se traduce en que, desde Home, solo podemos transitar a las páginas: Best-seller, New Products, Order Inquiry, Search Request y Shopping Cart.

Con estas tablas y aplicando una regla de tres se han obtenido los porcentajes necesarios para cada tipo de usuario (Shopping, Browsing, Ordering) en base al 100%.

Lo siguiente es evaluar las navegaciones sobre la página web de Prestashop para obtener las diferentes URL's de las navegaciones posibles. Esto se ha realizado mediante el uso del explorador web Firefox y la opción de desarrollador de red. Una vez se pueden obtener y visualizar datos de las peticiones HTTP, lo que se ha de hecho ha sido realizar navegaciones a través del explorador web como si de un usuario normal se tratase. Con las tablas anteriores, se han ido siguiendo los pasos para conseguir las direcciones URL correspondientes a las 14 páginas que conforman TPC-W.

Según las reglas de implementación de la especificación de TPC-W [15], se han de cumplir una serie de requisitos. Estos son:

Cuando un usuario empieza una sesión se deben de contemplar 2 escenarios:

El usuario es un usuario no registrado y por lo tanto no es cliente. Este escenario se produce un 20% de las veces y además el visitante puede convertirse en usuario registrado en cualquier momento.

La petición la realiza un usuario que es cliente ya registrado. Este escenario se da un 80% de las veces, además un usuario que empieza como visitante no puede logearse más tarde como cliente ya registrado.

En cualquier momento en el que se puede transitar a la página del carrito de compra (Shopping Cart), se ha de cumplir que, en caso de tener el carro de compra vacío se añade un producto promocional al azar. De esta forma, se evita que se transite a algún estado que solo se puede alcanzar cuando el carrito de compra contiene algún artículo. De hecho, esto nos dio varios errores en la implementación de la navegación hasta que se averiguo que era por este motivo.

En cuanto a la navegación que nos lleva al Order Inquiry, en cuya página necesitamos estar logueados para acceder a nuestro historial de pedidos, en caso de no estar logueado el usuario y ser un visitante se genera un login temporal con un usuario registrado al azar para poder alcanzar la página del Order Display. Una vez se transita a otro estado se produce un deslogueo del usuario para seguir navegando como visitante en el resto de las interacciones. Esto es una restricción de la especificación.

Para la creación de las navegaciones, se utiliza lo referente que se puede ver en la especificación de TPC-W, apartado 2 entero [15]. En este apartado, se especifican para cada página de las 14 que conforman TPC-W:

- Los requerimientos de entrada.
- La definición del proceso (como se comporta la página).
- La definición de la página de respuesta.
- Y las posibles navegaciones de los EB's.

Para todas las navegaciones que se han realizado, se han seguido lo más al pie de la letra dichos requerimientos, cumpliendo con todos aquellos que se especifican como obligatorios (must).

Ahora se detalla cómo se ha implementado la navegación para cada página que compone la web de TPC-W.

Inicialmente, en el estándar de TPC-W, se nos indica que un 80% de usuarios se loquean y un 20% navegan como visitantes de la web.

Aunque no es una navegación de TPC-W y no se contabiliza como tal, es necesario, para que los usuarios puedan loguearse, un estado y una función en el archivo de TGI TPCWUtilAction.java para hacer el registro. A esta función se le ha llamado GET\_EMAIL\_LOGIN. Esta función lo que hace es leer un fichero que se carga en un ArrayList con las 28.800 direcciones de correo electrónico pertenecientes a los clientes registrados. De esta manera, se elige con un Random una dirección al azar para realizar el logueo del usuario. Todas las funciones que se crean van en el archivo java mencionado anteriormente.

Por otra parte, la única forma de añadir artículos al carrito de compra es pasando por la página de Product Detail. Lo que se ha hecho ha sido comprobar si se viene de la página Product Detail y en este caso se añade el artículo al carrito de compra. En caso de que el artículo ya este, se incrementa en 1 la cantidad del artículo en el carrito.

También se han creado una serie de variables en el archivo de navegación XML, comunes a todas las transiciones. Estas variables son 3:

- Var\_login: Nos indica si un usuario está o no logueado en todo momento. Cuando se produce el logueo de un usuario esta variable cambia su valor a "yes".
- Empty\_cart: Nos indica si el carrito está vacío o contiene algún artículo.
- Donde\_vengo: Esta variable nos dice cuál es el estado anterior desde el que transitamos.

Para las páginas de HOME, BEST-SELLERS, NEW PRODUCTS y SEARCH REQUEST, la navegación, es simplemente acceder a la URL correspondiente ya que no tiene ninguna acción asociada.

ADMIN REQUEST Y ADMIN CONFIRM son tareas administrativas por lo que se accede al panel de control de Prestashop y se realiza alguna acción administrativa.

ORDER INQUIRY, si el usuario no está logueado, se accede a la página de logueo del usuario. Si está registrado el usuario, se accede a su espacio personal.

En ORDER DISPLAY, es dónde si el usuario no está logueado se hace uso de la función GET\_EMAIL\_LOGIN, y se hace un logueo momentáneo. Entonces, es cuando se consulta el historial de pedidos y una vez se muestra esta página, se desloguea el usuario otra vez para seguir navegando como invitado. Para los usuarios que ya están registrados simplemente se muestra el historial de pedidos.

SEARCH RESULTS se encarga de realizar la búsqueda y devolver los resultados. Su implementación, se ha llevado a cabo mediante una función llamada GET\_SEARCH\_QUERY. Este método, lo que primero que hace es generar de forma aleatoria el método de búsqueda, para elegir si la hace por autor, título o categoría. Después genera un nombre, apellido, título o categoría de forma aleatoria y se la devuelve al archivo XML que la utiliza en su URL para poder realizar la acción de búsqueda.

Para PRODCUT DETAIL se ha tenido que añadir otra función, GET\_ID\_PRODUCT, que se encarga de elegir el producto del cual se quieren detalles. Esto se ha conseguido almacenando en una variable el código html de la página anterior. En los estados anteriores posibles que permiten alcanzar el de Product Detail se encuentra en su código los diferentes identificadores de productos. De esta forma, el método busca estos identificadores y los va almacenando en un Array para luego elegir uno al azar y se lo pasa al XML. Este es el identificador del producto que se pasa en la URL de la petición. A continuación, se detallan los estados que componen todos ellos una acción de compra, si se ejecutan todos de cierta manera.

SHOPPING CART ha sido uno de los más complicados de realizar ya que tiene bastantes cosas que implementar. Lo primero es ver si se transita del estado Product Detail, si así es, se ha de añadir el artículo al carro de compra. Para ello se han creado varios métodos que son necesarios: GET\_ID\_PRODUCT\_CART, GET\_TOKEN\_CART\_SHOPPING y GET\_RAND\_EPOCH. El primero de ellos se encarga de coger el identificador del producto que queremos elegir, esta información está en el html de Product Detail. Además, hace falta un token que es único para cada usuario y el cual se puede encontrar en el código html también. El último de estos métodos se encarga de generar la fecha actual en formato Epoch. Una vez se han obtenido estos valores, se crea la URL que nos permite añadir el producto al carro de compra y se generan los valores para realizar la petición POST.

Como segunda parte de esta navegación, es necesario, como se ha comentado anteriormente, el añadir un artículo promocional al carro de compra en caso de estar vacío. De este fin se encarga la función IS\_CART\_EMPTY, la cual nos dice si está vacío o no el carro, cosa que sabemos mirando el código html de la página actual. Para el caso de que el carro este vacío se han creado una función más, GET\_TOKEN\_CART\_SHOPPING, la cual utiliza para recoger el token que puede haber cambiado. Con este valor y con el de la fecha ya se puede realizar la petición POST correspondiente.

Una vez se han comprobado los pasos anteriores ya se puede transitar a la página del usuario del carrito.

CUSTOMER REGISTRATION es la navegación que más funciones necesita. Si el usuario está logueado, no hay ningún problema en la navegación, pero si el usuario no está registrado, son necesarios los datos del cliente como el nombre, apellidos, dirección, país, provincia, teléfono, etc. Además, para poder acceder a la URL de la petición POST hace falta recoger el token de nuevo, generar una fecha en formato Epoch y la dirección de email a registrar. De esta forma, lo primero que se hace es recoger el token, la fecha y generar una dirección de correo aleatoria para crear el usuario. Esto se ha conseguido con las funciones GET\_REGISTER\_EMAIL y GET\_TOKEN\_REGISTER. Para la fecha se hace uso del método descrito anteriormente. La primera función crea una dirección a partir de los ficheros de nombres y apellidos, eligiéndolos al azar y concatenándolos para luego poner la terminación de direcciones que esta almacenada en un Array al igual que en TPC-W. Este nombre y apellido se guarda para luego usarlo en el formulario de registro de usuario. Una vez se realiza este paso, viene el formulario para añadir los datos de contacto con los métodos GET\_REGISTER\_FIRSTNAME, GET\_REGISTER\_LASTNAME. De esta forma se ha podido realizar la petición POST del formulario de registro. A partir de este momento se cambia la variable var\_login a yes.

Después del proceso anterior se han de generar los datos del usuario para su registro mediante los métodos:

```
GET_REGISTER_ADDRESS
GET_REGISTER_ZIP
GET_REGISTER_CITY
GET_REGISTER_COUNTRY
GET_REGISTER_MOBILE_PHONE
GET_REGISTER_STATE
GET_TOKEN_REGISTER
```

Dada la obviedad de sus nombres, no se va a explicar para que sirve cada uno. Con todo esto ya tenemos la navegación del registro de usuario terminada.

En BUY REQUEST lo primero que se ha de conseguir es la dirección de envío del cliente. Este dato se obtiene del html mediante la función GET\_ID\_ADDRESS\_CUST\_REGIST2 y se realiza una petición POST pasando este parámetro. La siguiente acción que se puede realizar es elegir el método de envío de entre dos posibilidades, esto se ha generado con una función que devuelve un valor aleatorio de entre los dos posibles (1 o 2), GET\_SHIPPING\_MODE. En este punto hay que decir que, si se elige una forma de envío u otra se genera un Post que ejecuta Prestashop sin intervención del usuario, por lo que se ha tenido que implementar este POST si se elige el envío con valor 2, porque el 1 es por defecto y no genera este POST. Para finalizar se

ejecutan un par de POSTs más, el primero de ellos usa datos que ya se han recogido anteriormente y el último es el de método de pago por lo que se ha creado la función `GET_PAYMENT_MODE` que devuelve un valor aleatorio que corresponde con una forma de pago. Se han dejado los 2 métodos de pago que trae Prestashop por defecto que son por cheque (cheque) o por transferencia bancaria (bankwire). Si se añaden nuevos módulos con más métodos de pago habrá que modificar el Array que contiene estos valores y añadir las nuevas formas de pago.

Para finalizar, el último estado que nos queda por definir es el de `BUY_CONFIRM`, el cual tiene su complicación ya que es necesario acceder a la base de datos de Prestashop para extraer unos valores que son necesarios para realizar la confirmación del pedido. `GET_ID_CART` se crea para saber que número de carro de compra tiene asociado el usuario ya que este se crea en la base de datos en el momento de añadir algún artículo al carro.

`GET_SECURE_KEY` es el encargado de extraer la clave de seguridad que utiliza Prestashop para los usuarios cuando realizan las compras. Es una clave distinta y única para cada usuario. El mismo usuario siempre tiene la misma clave, que se crea y asocia cuando el usuario se registra en la página web.

`GET_ID_ORDER` lo que hace es averiguar cuál es el identificador del último pedido realizado en la base de datos y de esta forma el nuevo pedido se crea con este valor +1. La confirmación del pedido no se realiza mediante un método POST, sino que se hace con una petición GET pasando estos valores en la URL.

Hay que remarcar que, estos métodos al poder ser accedidos desde múltiples hilos de ejecución se han tenido que hacer sincronizados mediante la instrucción necesaria en Java (`synchronized`), de esta manera, no se pueden producir accesos simultáneos que darían resultados erróneos, como por ejemplo, identificadores de pedidos no válidos. Se ha comprobado que, si los métodos no están sincronizados y tenemos muchos usuarios, realmente no se realizan bien los pedidos.

### **5.3. Validación de las navegaciones**

En este apartado se ejecuta una serie de veces GUERNICA para comprobar que se han realizado las navegaciones de forma correcta. La forma de hacer la validación se hace mediante el panel de control de Prestashop, para ver si se registran los usuarios, hacen búsquedas bien los visitantes, se loguean, hacen compras válidas y todas las acciones que hayamos configurado en la navegación. Esto se hace observando los nombres y apellidos de los nuevos clientes, si aumenta el número de estos, al igual que el de los pedidos, los carritos de compra, la forma de pago de los pedidos y se hace uso de la opción de visitantes en línea para ver quien está en la web en tiempo real. También en este punto, se ha utilizado la consola de eclipse para mostrar valores que luego son usados por Prestashop, como nombres, apellidos, dirección de envío, tipo de pago elegido, identificador de producto elegido, etc. y de esta forma validar los datos de forma

Adaptación de un Benchmark para e-commerce bajo plataformas actuales de ventas como Prestashop haciendo uso del generador de carga GUERNICA

rápida. Por ejemplo, de esta manera, se puede ver rápidamente cuando se crea un usuario y si lo hace de forma correcta.

Tras realizar varios experimentos con diferentes usuarios, se han obtenido los siguientes resultados en el backend de Prestashop:

The image shows two tables from the Prestashop backend. The first table, titled 'Clientes conectados en este momento', lists 9 active clients with their IDs and names. The second table, titled 'Visitantes conectados en este momento', lists 1 active visitor with their ID, IP address, and last activity time.

Clientes conectados en este momento	
Total: 9	
ID del cliente	Nombre
12680	Abigail Malburgo
22579	Adi Shankara Goya
28853	Amira Ariza
2443	Archibaldo Illan
28851	Karmele Estrella
17051	Menaka Calon
26958	Olinta Ramales
2285	Rosario Pedronera
28852	Slavko Campo

Visitantes conectados en este momento		
Total: 1		
ID de invitado	IP	Última actividad
653	127.0.0.1	21:50:51

Figura 33: Validación de los 10 usuarios conectados Prestashop

En la figura 33 se puede observar que lanzando un experimento de 10 usuarios estos se pueden ver a través del panel de control de Prestashop. Para 20 usuarios se aprecia en la siguiente imagen:

Adaptación de un Benchmark para e-commerce bajo plataformas actuales de ventas como Prestashop haciendo uso del generador de carga GUERNICA

Clientes conectados en este momento

Total: 20

ID del cliente	Nombre
16138	Albina Zaldo
13613	Amilcar Biembengen
2390	Brunilda Pons
21647	Celia Ulloa
3525	Devajuti Plo
5725	Drona Alzorbe
22479	Fortunato Valcazar
26064	Genoveva Zuriarrain
1466	Kushika Juanes
9679	Mabel Almandoz
28860	Marcelo Villarroel
28859	Marcelo Villarroel
13340	Mariano Usarralde
18910	Melisa Sanjuan
23634	Menaka Ribeiro
11505	Mirtha Loinaz
27077	Rishikesh Olier
17465	Sheila Olea
28858	Valentina Saenz
10442	Wenceslao Panecatl

Visitantes conectados en este momento

Figura 34: Validación de los 20 usuarios conectados en Prestashop

Obteniendo los mismos resultados para 30 usuarios. Además, podemos ver en la figura 35 y en la figura 36 como se registran nuevos usuarios y se realizan nuevos pedidos.



Figura 35: Validación de registro de usuarios en Prestashop

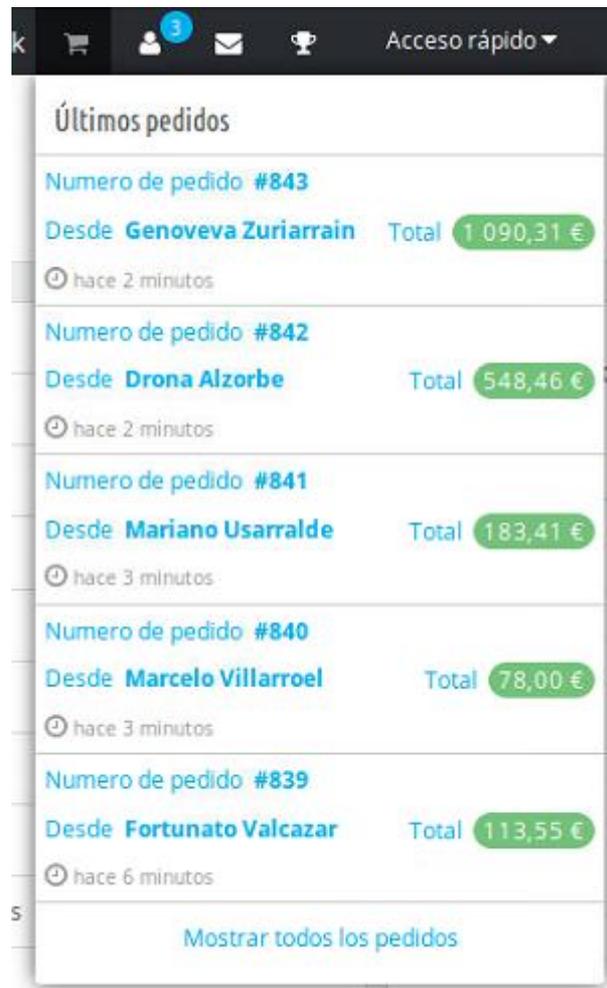
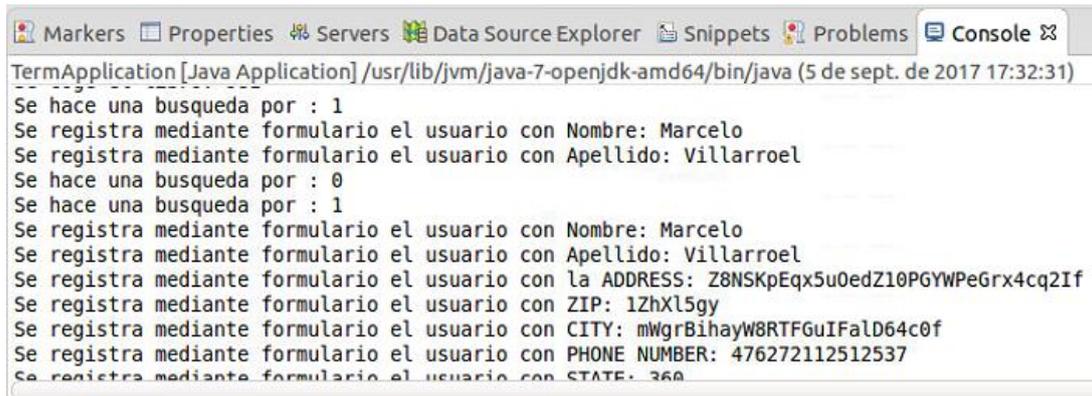


Figura 36: Validación pedidos realizados Prestashop

También haciendo uso del archivo de debug que genera GUERNICA, se puede comprobar que métodos se han ejecutado, que valores se han obtenido, desde que página se ha transitado, cuál ha sido la palabra usada para hacer una búsqueda, si ha sido por autor, título o categoría, etc. De esta forma, se comprueban también estos valores que podemos observar en la consola de eclipse y ver que son correctos, cosa que se ha comprobado de forma satisfactoria como se aprecia en la siguiente imagen:

Adaptación de un Benchmark para e-commerce bajo plataformas actuales de ventas como Prestashop haciendo uso del generador de carga GUERNICA



```
TermApplication [Java Application] /usr/lib/jvm/java-7-openjdk-amd64/bin/java (5 de sept. de 2017 17:32:31)
Se hace una búsqueda por : 1
Se registra mediante formulario el usuario con Nombre: Marcelo
Se registra mediante formulario el usuario con Apellido: Villarroel
Se hace una búsqueda por : 0
Se hace una búsqueda por : 1
Se registra mediante formulario el usuario con Nombre: Marcelo
Se registra mediante formulario el usuario con Apellido: Villarroel
Se registra mediante formulario el usuario con la ADDRESS: Z8NSKpEqx5u0edZ10PGYWPeGrx4cq2If
Se registra mediante formulario el usuario con ZIP: 1ZhXl5gy
Se registra mediante formulario el usuario con CITY: mWgrBihayW8RTFGuIFalD64c0f
Se registra mediante formulario el usuario con PHONE NUMBER: 476272112512537
Se registra mediante formulario el usuario con STATE: 360
```

Figura 37: Validación haciendo uso de la consola de Eclipse

Con todo lo anteriormente expuesto se puede decir que, efectivamente, las navegaciones son correctas y se realizan todas las acciones especificadas sobre la web de forma válida. En este punto del proyecto, se consigue tener un modelo de navegación óptimo para Prestashop, bajo los diferentes tipos de usuarios especificados por TPC-W y generando la carga de forma dinámica con GUERNICA.

## 6 Experimentos

---

En el siguiente capítulo, se van a mostrar los diferentes tipos de experimentos realizados y las opciones establecidas para cada uno de ellos. Además, se expondrán los datos recolectados, los resultados obtenidos y la validación de estos respecto al estándar TPC-W.

### 6.1. Descripción de los experimentos

Se hace uso de los experimentos realizados en el capítulo 5 apartado 3 para la validación de los cambios efectuados en TPC-W, de esta manera se tienen resultados para 10, 20 y 30 usuarios de cada tipo de usuario (Browsing, Shopping y Ordering) para la plataforma de TPC-W. De estos resultados nos interesan el número de peticiones que se producen de cada página web.

Se generan los mismos experimentos para Prestashop haciendo uso del planificador de GUERNICA para cada tipo de usuario también. Los resultados que nos interesan son los mismos que en TPC-W, el número de peticiones que se producen de cada página.

Ambos experimentos se realizan durante 30 minutos de tiempo total con 2 minutos de calentamiento y 2 de rampa de bajada, suficiente para simular una navegación real en cualquier página web hoy en día. Y todos ellos se realizan con una base de datos con factor de escala de 1.000 artículos.

Cabe mencionar que estos experimentos se han realizado 10 veces, para comprobar que no se producían desfases no aceptables en los resultados obtenidos, logrando resultados parejos entre todos ellos. En el siguiente capítulo se exponen los resultados obtenidos de los últimos experimentos realizados tanto para TPC-W como para Prestashop.

Los tiempos de pensar son iguales en todos los casos. Corresponden a un valor de 7 segundos con un 1 segundo, más menos, de margen. Entre las acciones de Prestashop sin interacción por parte del usuario este tiempo equivale a 0.

### 6.2. Datos obtenidos

A continuación, se muestran una de las tablas con los correspondientes datos obtenidos para uno de los experimentos que se han realizado en el apartado anterior. En este caso, se trata de la tabla para el experimento de navegación Shopping con 10 usuarios.

EXPERIMENTO SHOPPING MIX 10 USUARIOS
--------------------------------------

PÁGINA VISTA	N.º DE VECES	PRESTASHOP (%)	TPC-W (%)
HOME	300	16,788	16
NEW PRODUCTS	95	5,3161	5
BEST SELLERS	96	5,3721	5
PRODUCT DETAIL	289	16,172	17
SEARCH REQUEST	335	18,746	20
SEARCH RESULT	274	15,333	17
SHOPPING CART	237	13,262	11,60
REGISTRATION	61	3,413	3
BUY REQUEST	56	3,133	2,60
BUY CONFIRM	24	1,3430	1,20
ORDER INQUIRY	10	0,5596	0,75
ORDER DISPLAY	10	0,5596	0,66
ADM. REQUEST	0	0	0,10
ADM. CONFIRM	0	0	0,09

Tabla 5: Resultados obtenidos para shopping mix con 10 usuarios en Prestashop

La interpretación de esta tabla es la siguiente:

La primera columna equivale a cada una de las 14 páginas que conforman TPC-W.

La segunda columna, muestra el número de veces que se ha visitado dicha página en la navegación. Para este valor, no se computan los ficheros .css, js o lo que pueda ser descargado de la página adicionalmente, como los objetos embebidos, sino que simplemente, cuenta las veces que se ha visitado dicha página. Es decir, solo se computa cuando se visita esta página sin importar las conexiones adicionales, por lo que el resultado total de visitas es el número total de veces que se han visitado dichas páginas.

La tercera columna equivale al porcentaje de visitas en tanto por cien que se ha conseguido en las navegaciones de Prestashop.

La última columna muestra el porcentaje que especifica TPC-W, en tanto por cien, que se puede observar en la figura 9.

Para los experimentos de 20 y 30 usuarios se han recogido tablas de similar forma a la anterior, aunque se omite la inclusión de estas en el documento por ser demasiado repetitivas.

De esta forma, se han obtenido los porcentajes para cada tipo de usuario con 3 tipos de cargas distintas de 10, 20 y 30 usuarios, como se hizo en los experimentos de TPC-W del capítulo 3 apartado 5.

### **6.3. Análisis de resultados**

Tomando como punto de partida los datos del apartado anterior, se han obtenido las gráficas correspondientes a cada experimento.

De esta manera se pretende ver si los porcentajes que se producen en Prestashop usando GUERNICA, son similares a los porcentajes de navegación que especifica la documentación de TPC-W.

Según TPC-W, todas las navegaciones se pueden dividir en 2 partes, una de navegación y otra de compras. TPC-W especifica que:

- Browsing Mix ha de tener un 95% de navegación (BROWSE) y un 5% de compras (ORDER).
- Shopping Mix ha de tener un 80% de navegación (BROWSE) y un 20% de compras (ORDER).
- Ordering Mix ha de tener un 50% de navegación (BROWSE) y un 50% de compras (ORDER).

Las gráficas que se muestran a continuación se han dividido en estas dos partes para poder apreciar mejor los resultados.

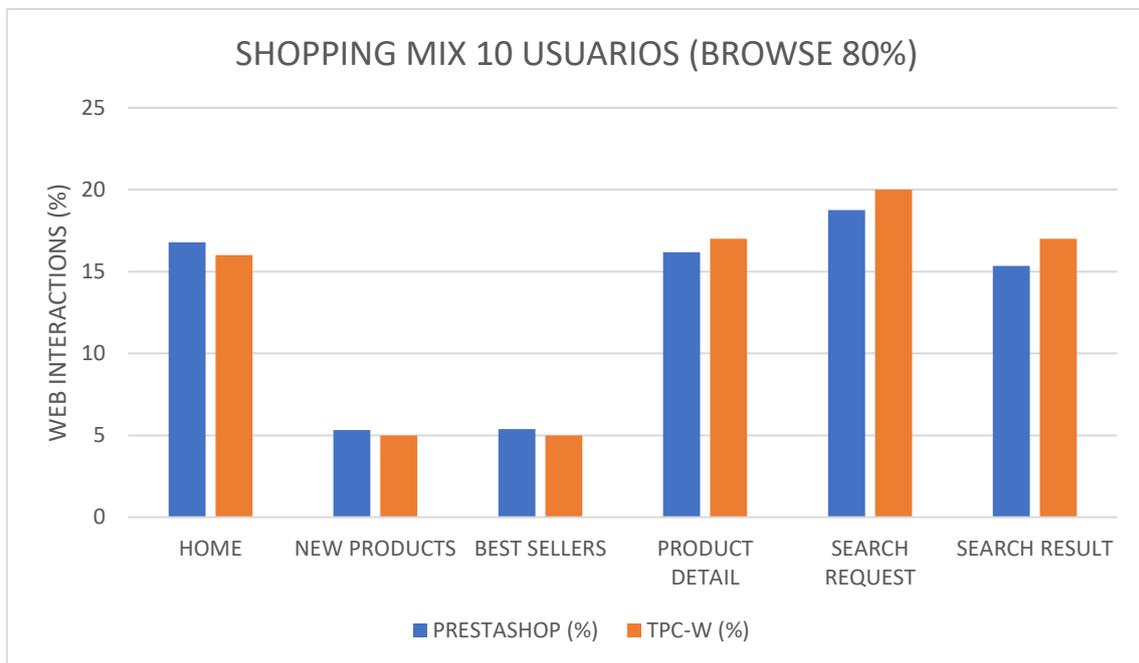


Figura 38: Shopping Mix 10 usuarios (BROWSE)

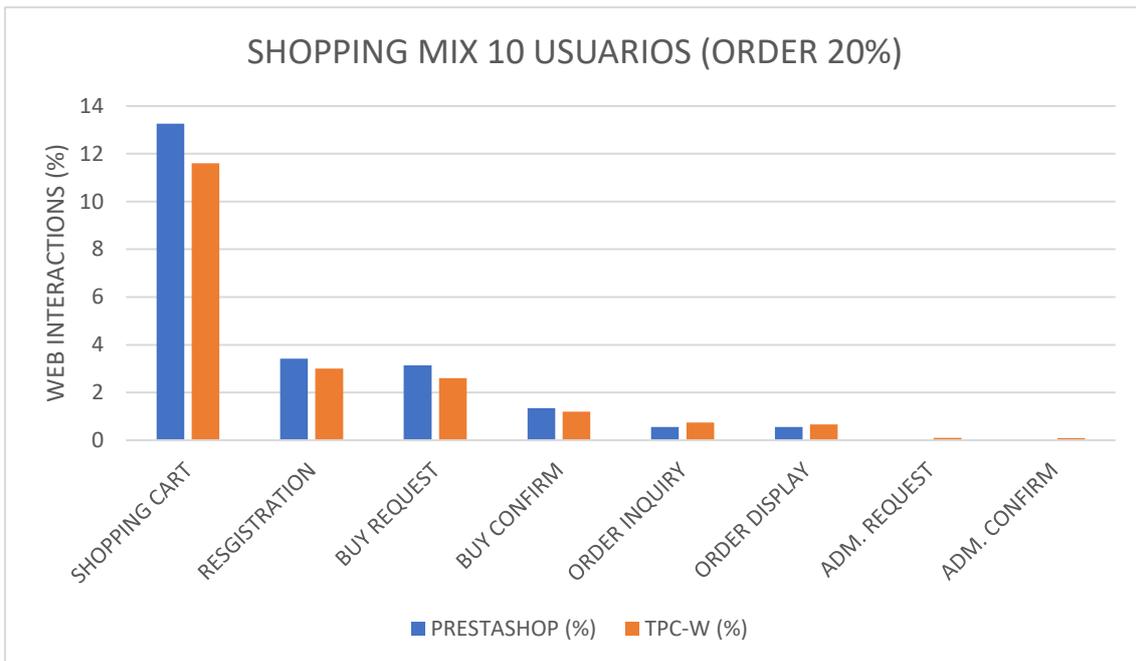


Figura 39: Shopping Mix 10 usuarios (ORDER)

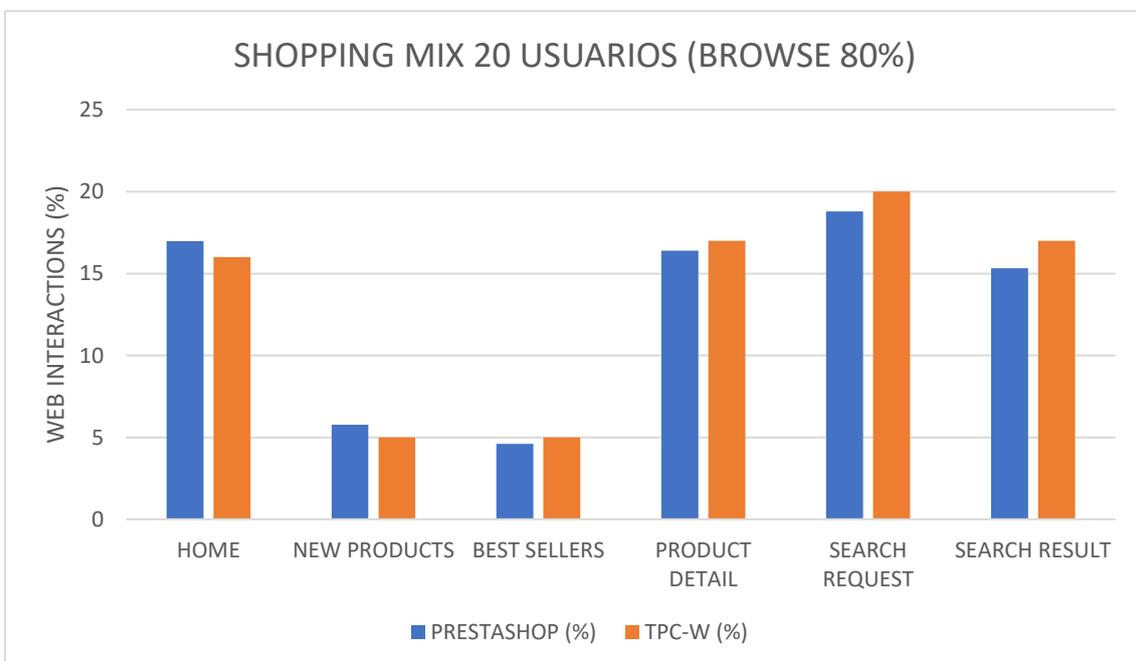


Figura 40: Shopping Mix 20 usuarios (BROWSER)

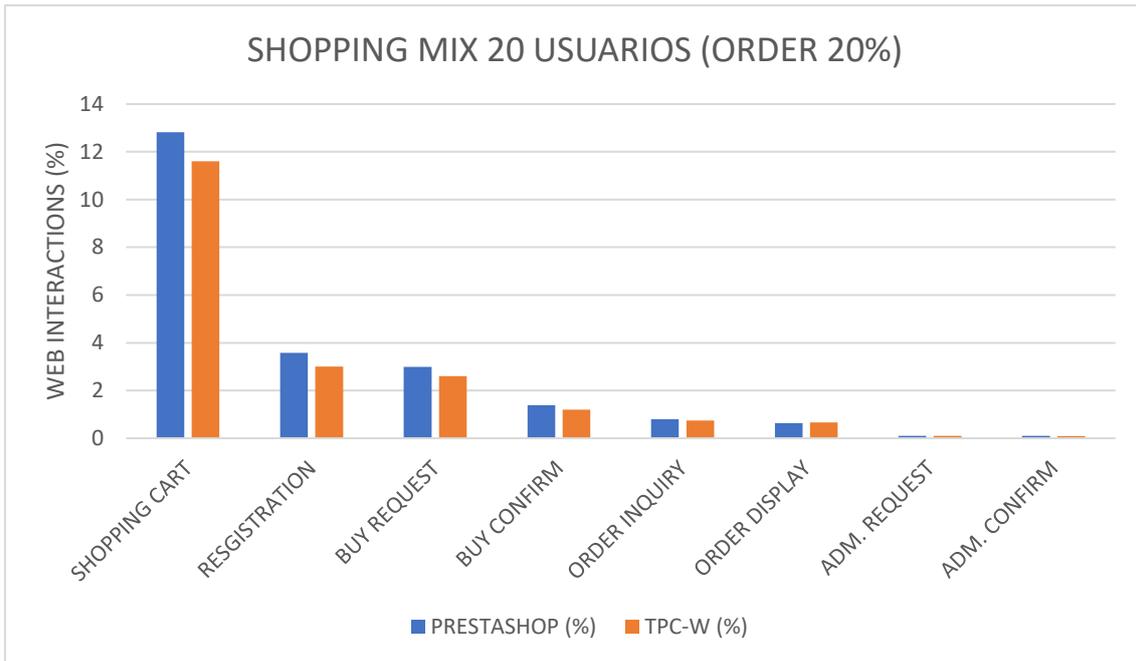


Figura 41: Shopping Mix 20 usuarios (ORDER)

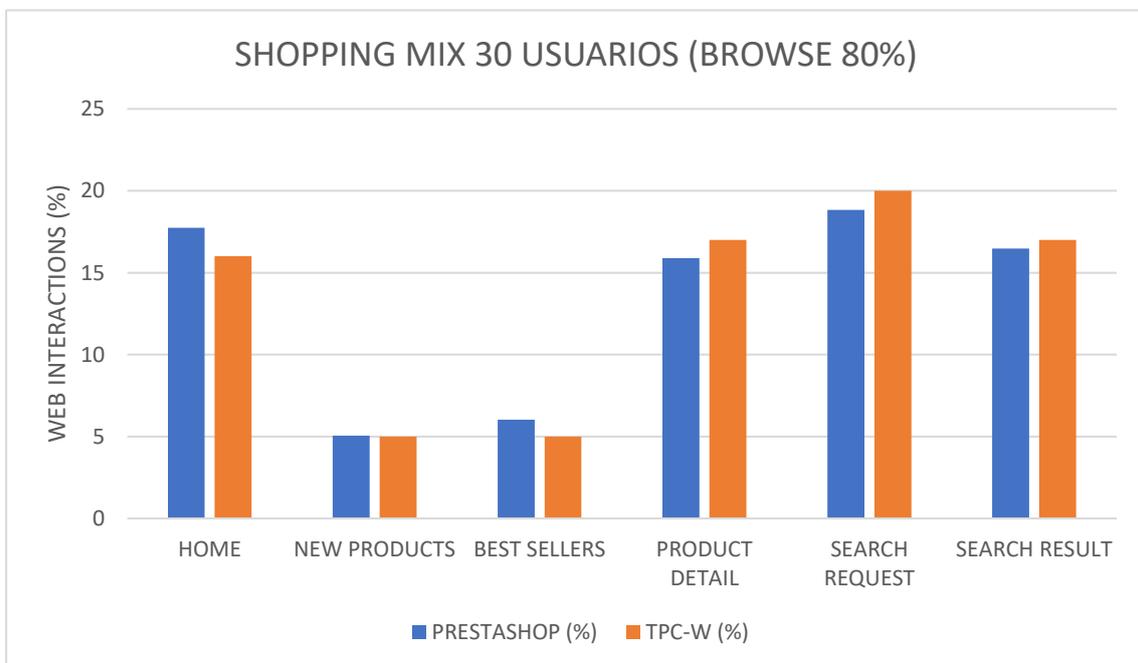


Figura 42: Shopping Mix 30 usuarios (BROWSER)

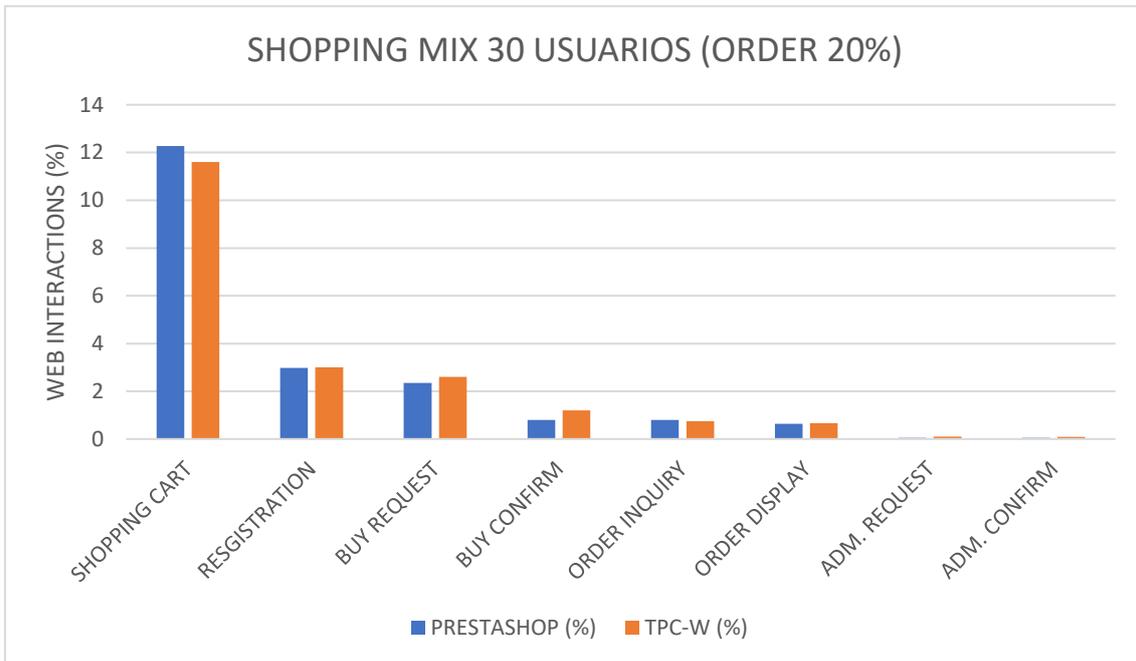


Figura 43: Shopping Mix 30 usuarios (ORDER)

A continuación, se muestran las gráficas para el Browsing.

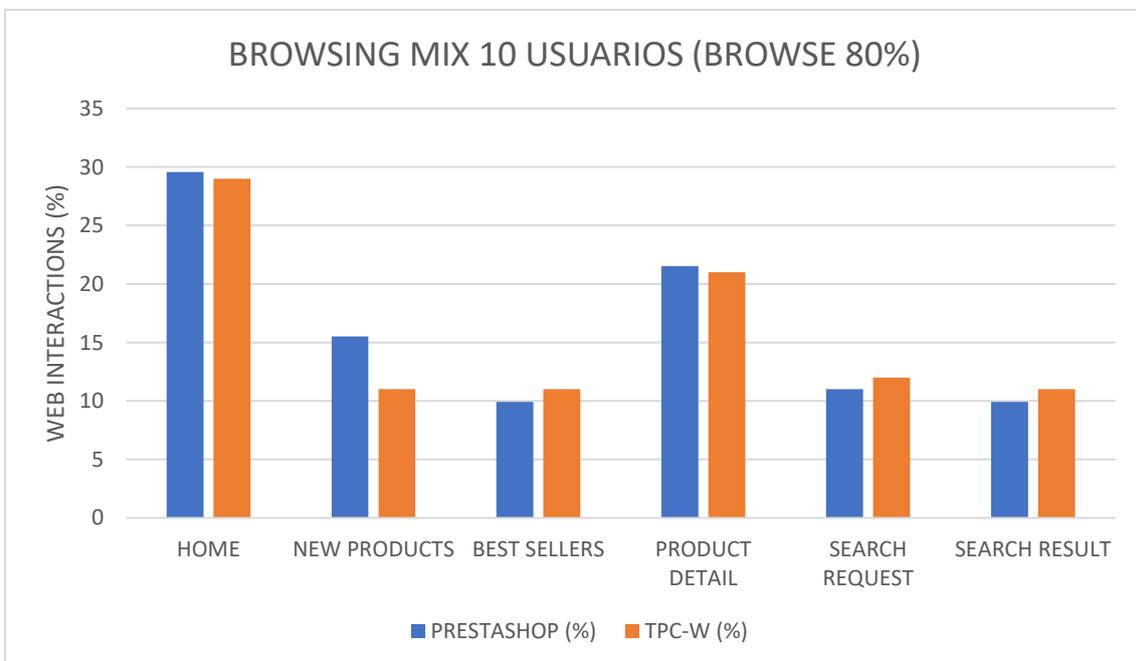


Figura 44: Browsing Mix 10 usuarios (BROWSER)

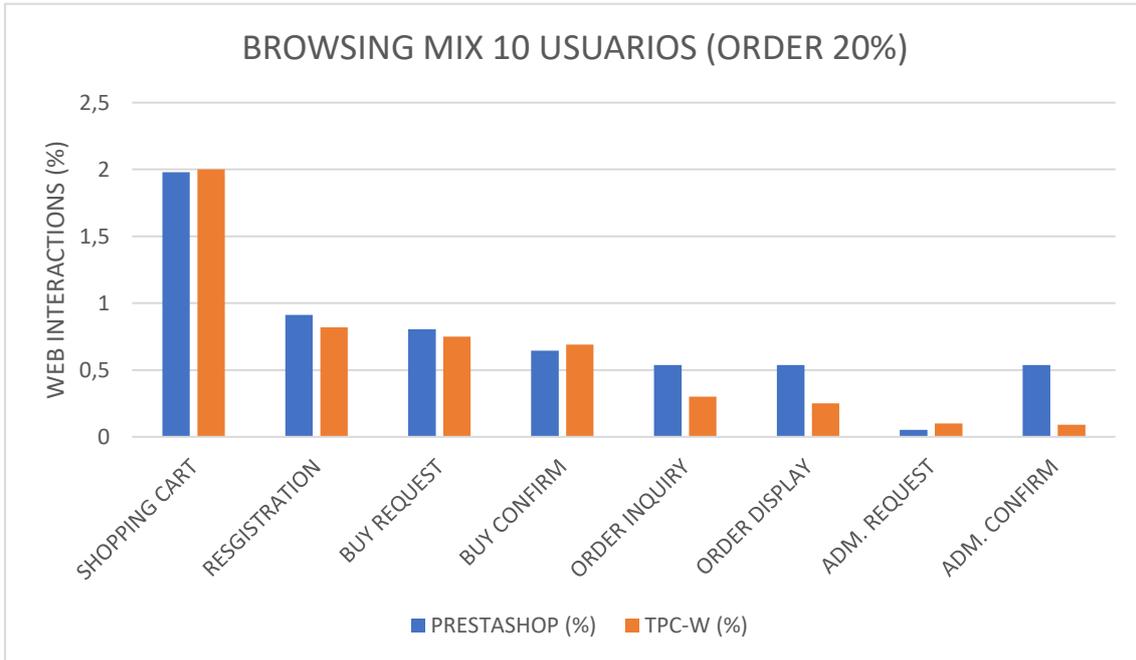


Figura 45: Browsing Mix 10 usuarios (ORDER)

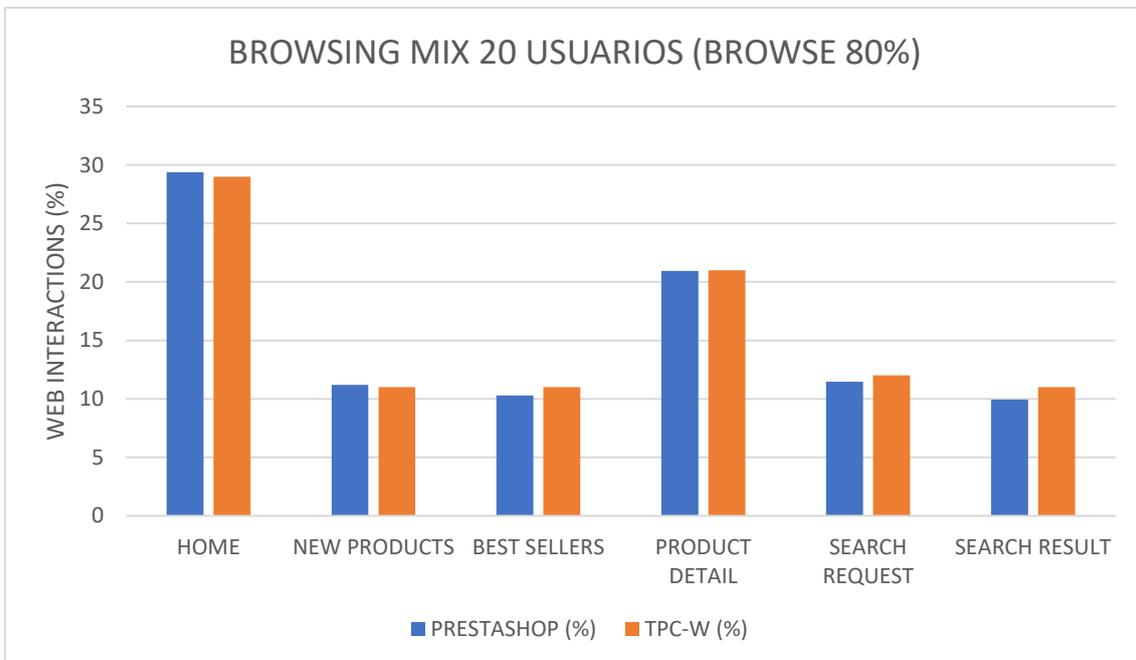


Figura 46: Browsing Mix 20 usuarios (BROWSER)

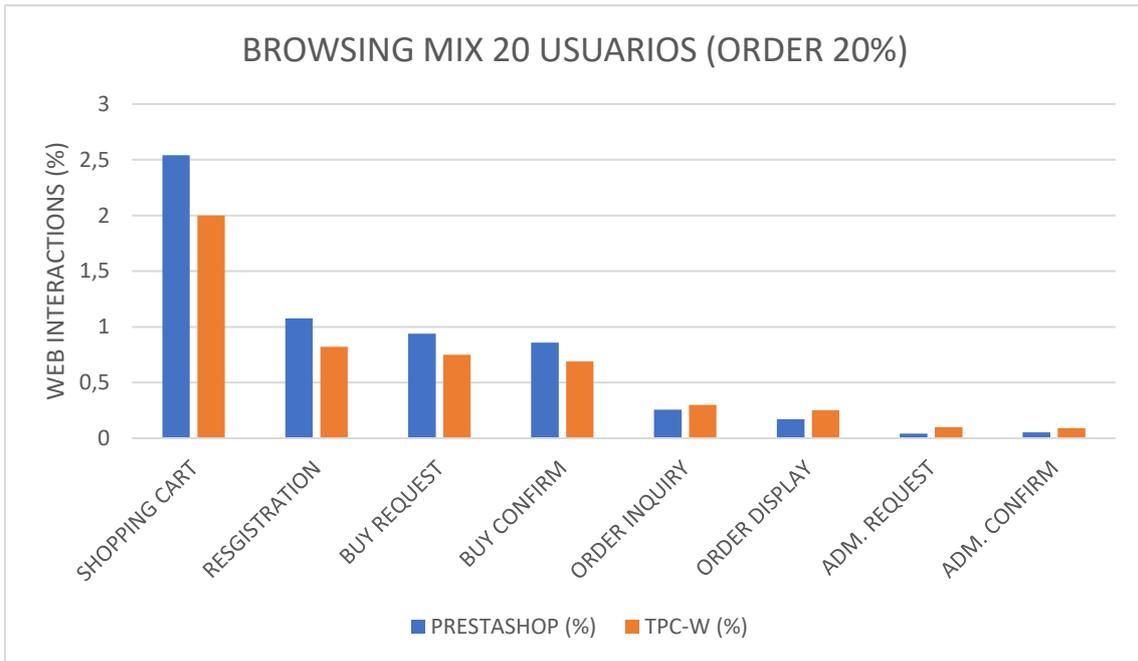


Figura 47: Browsing Mix 20 usuarios (ORDER)

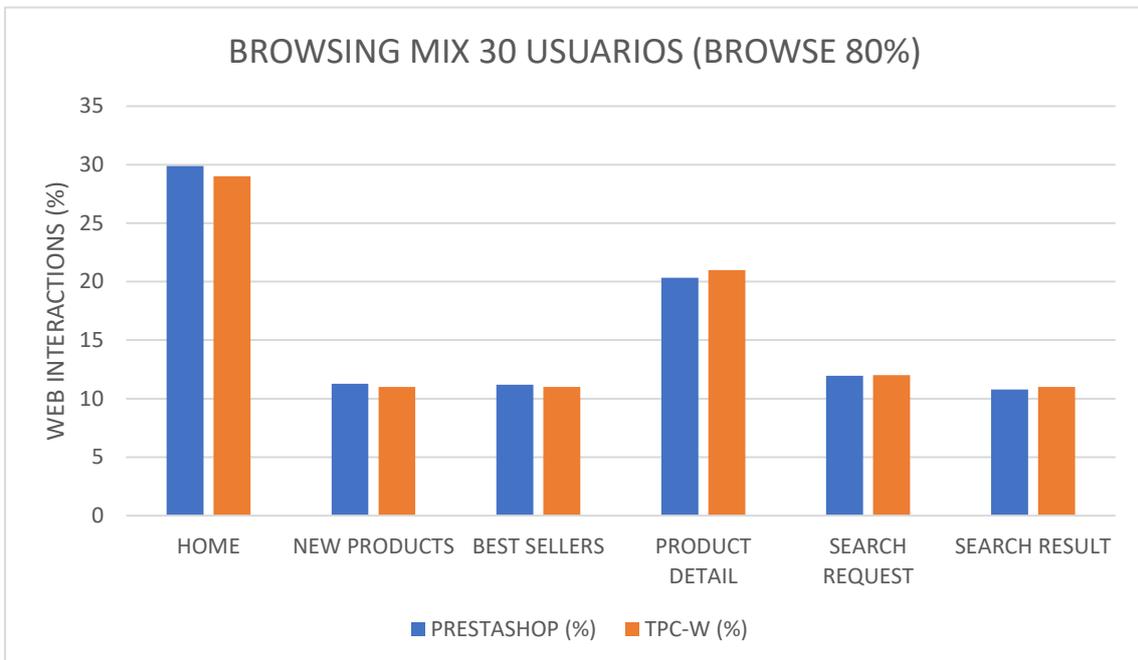


Figura 48: Browsing Mix 30 usuarios (BROWSER)

Adaptación de un Benchmark para e-commerce bajo plataformas actuales de ventas como Prestashop haciendo uso del generador de carga GUERNICA

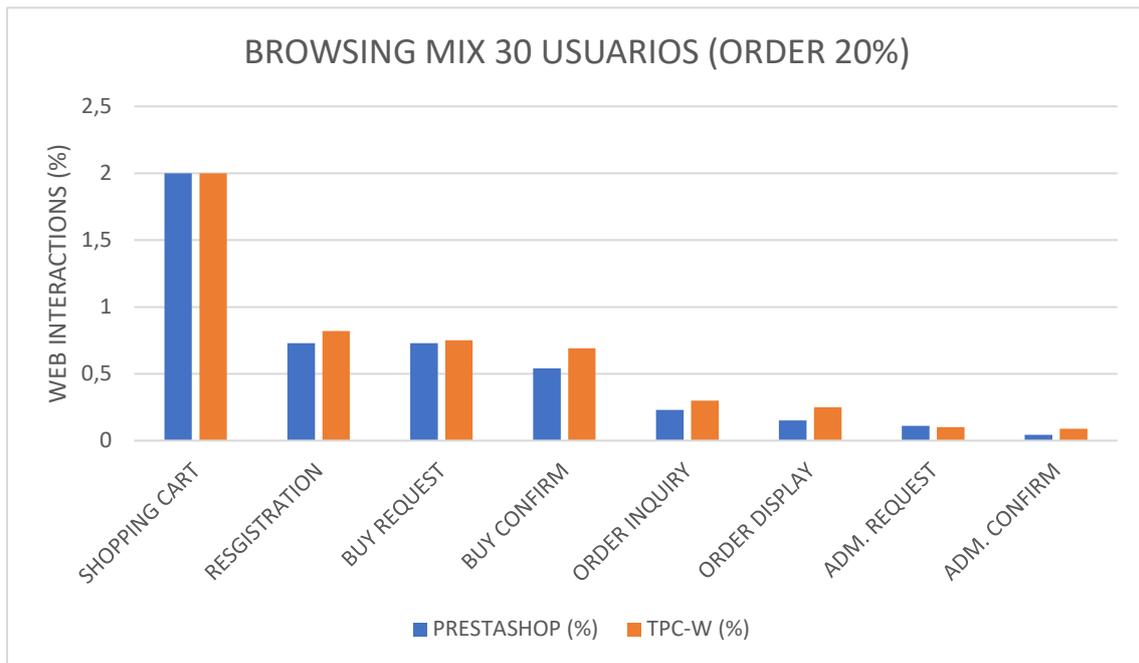


Figura 49: Browsing Mix 30 usuarios (ORDER)

Y por último se muestran las gráficas para el usuario de tipo Ordering.

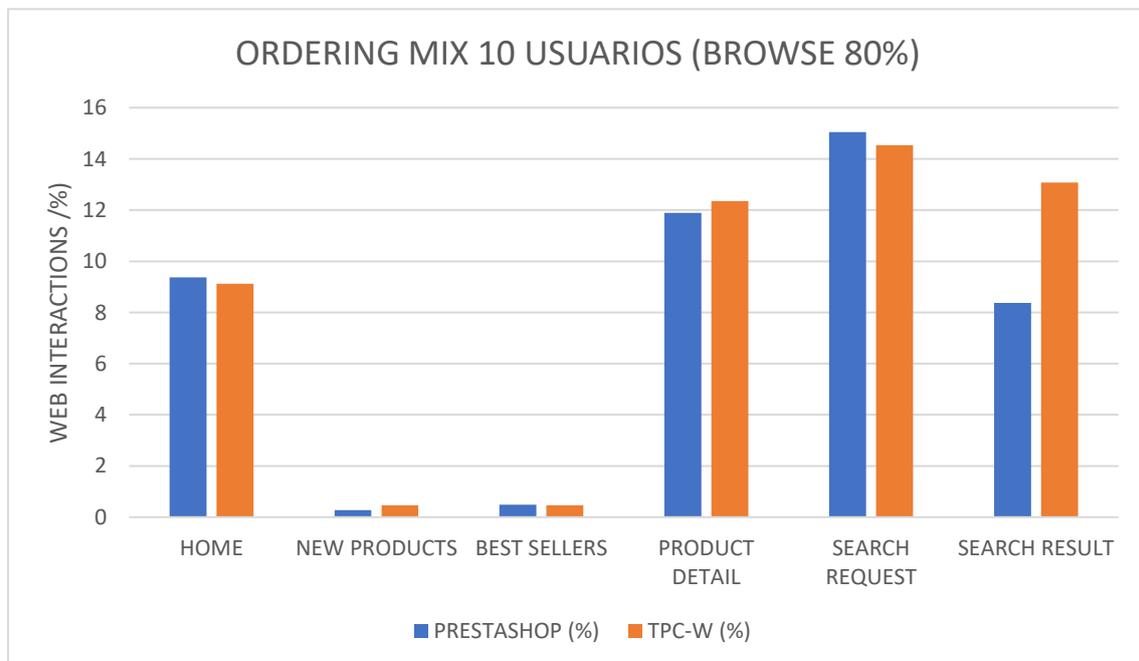


Figura 50: Ordering Mix 10 usuarios (BROWSER)

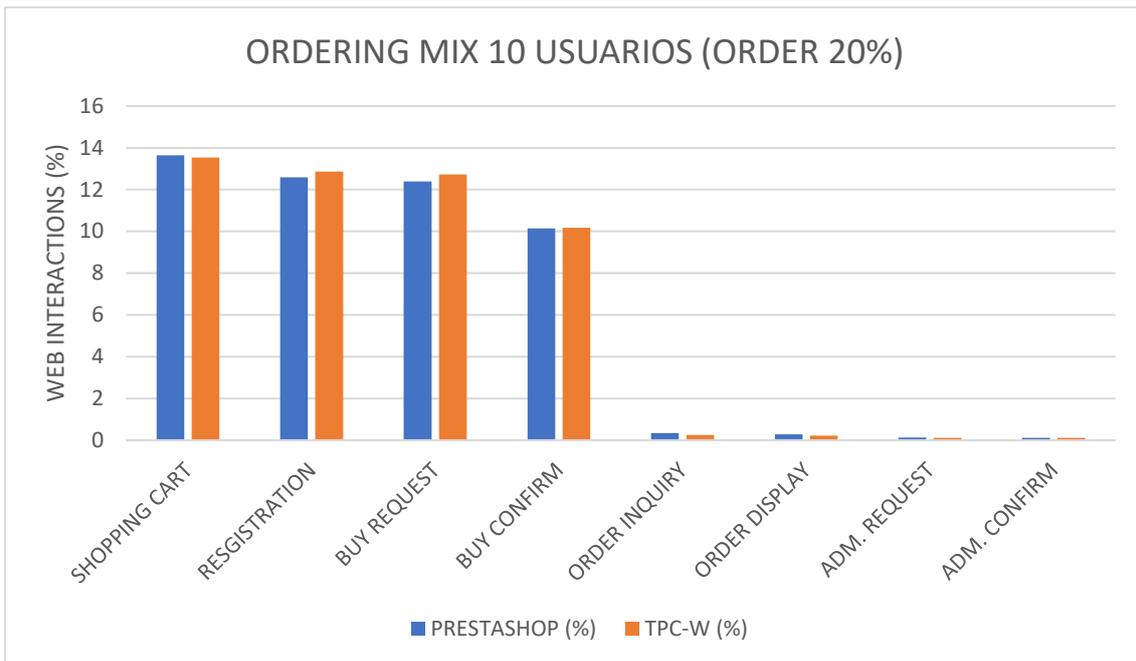


Figura 51: Ordering Mix 10 usuarios (ORDER)

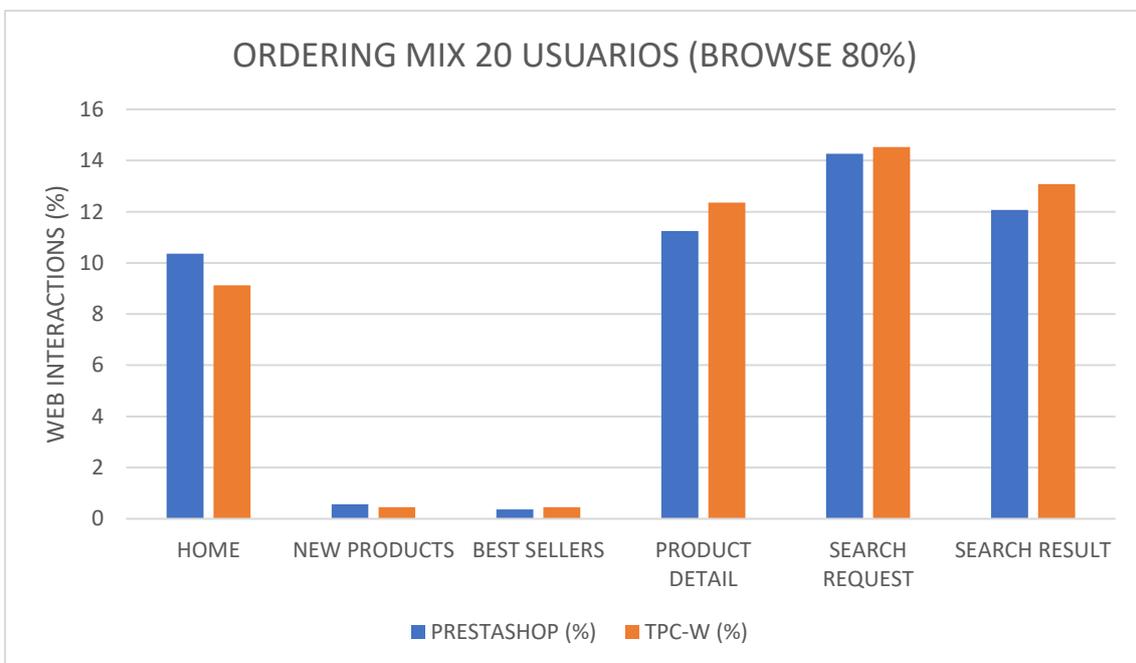


Figura 52: Ordering Mix 20 usuarios (BROWSE)

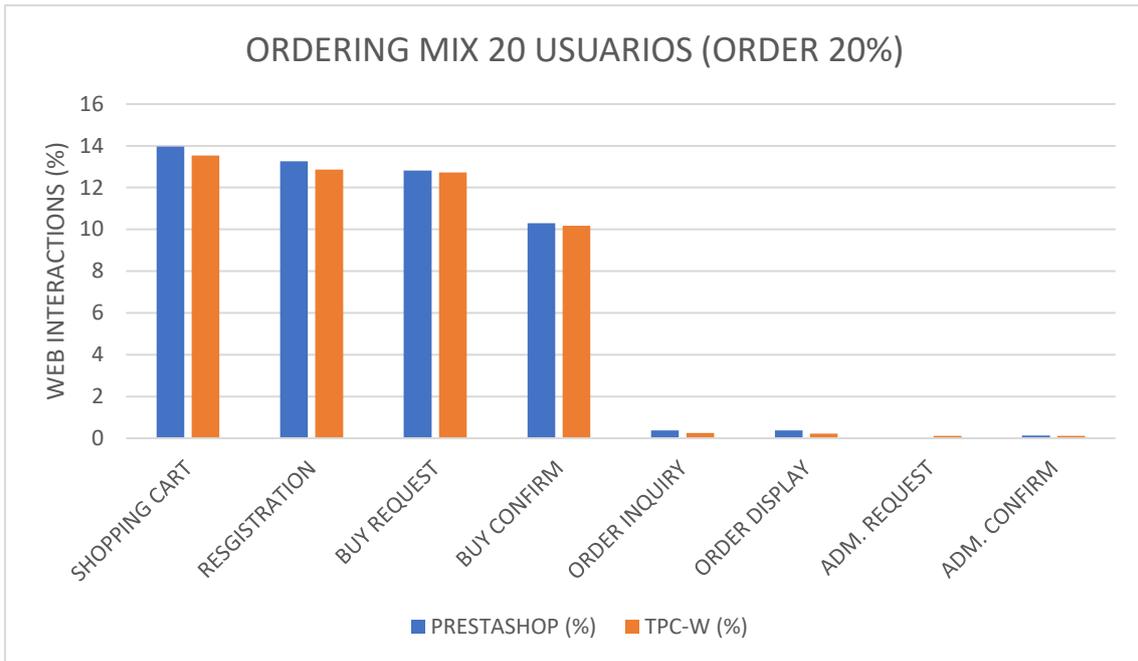


Figura 53: Ordering Mix 20 usuarios (ORDER)

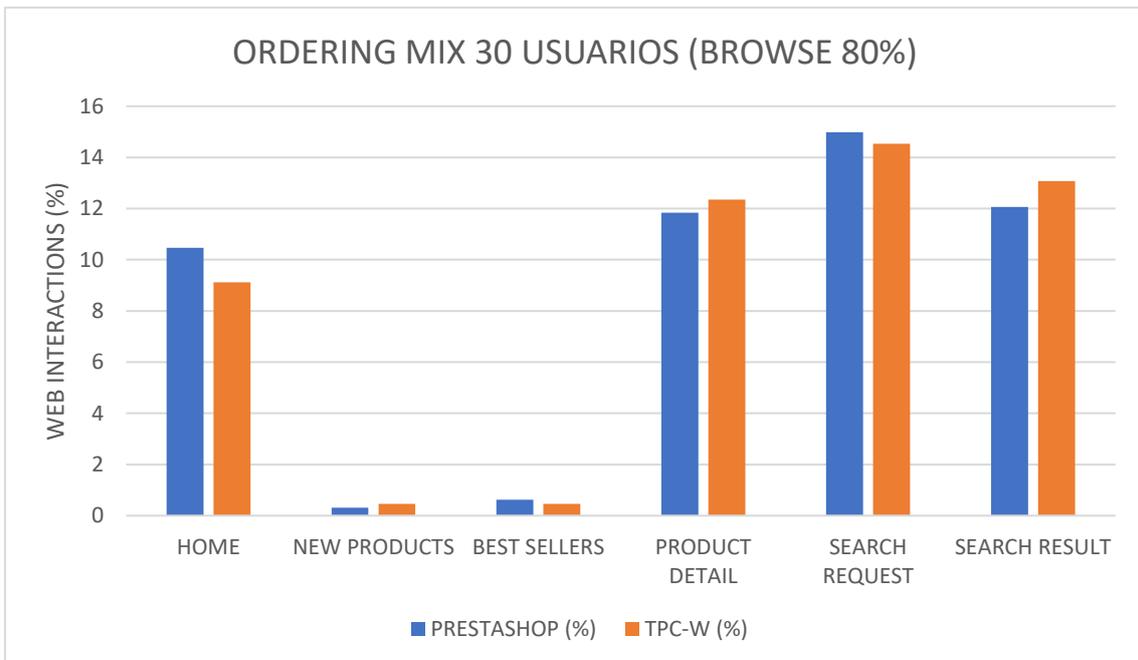


Figura 54: Ordering Mix 30 usuarios (BROWSER)

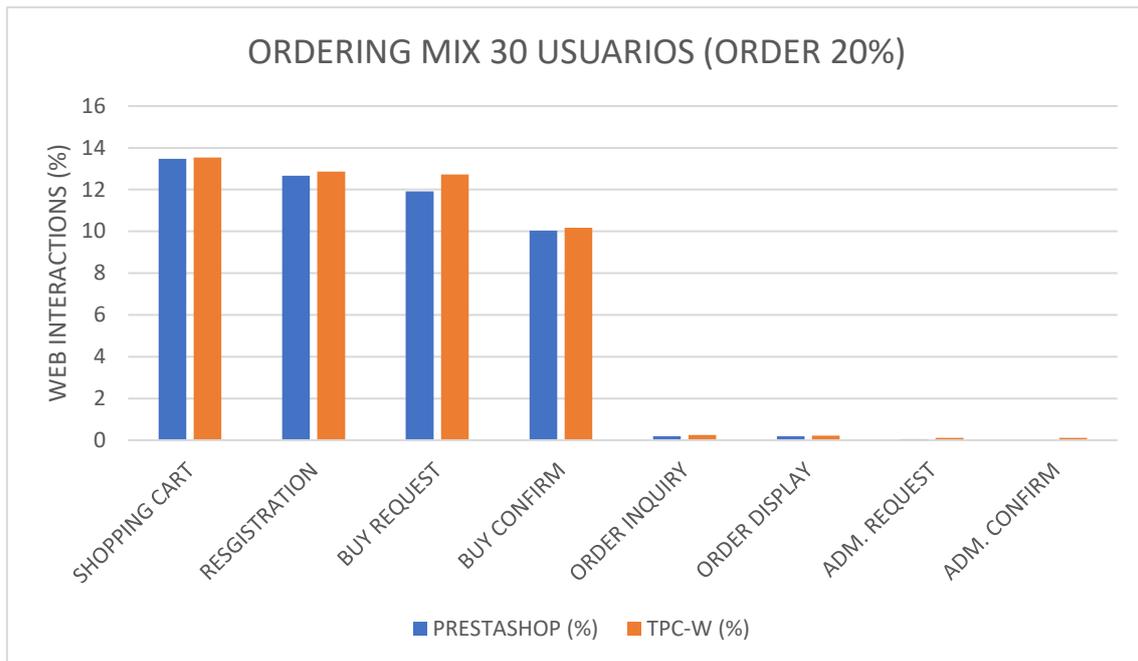


Figura 55: Ordering Mix 30 usuarios (ORDER)

Como se puede observar en las gráficas anteriores, se muestra que los porcentajes obtenidos mediante navegaciones de GUERNICA sobre la plataforma de comercio electrónico Prestashop, son prácticamente iguales a los que especifica TPC-W. Por lo que se puede confirmar que, efectivamente, tenemos navegaciones válidas generadas desde GUERNICA sobre la aplicación web de comercio electrónico Prestashop, y respetan el tipo de navegación para cada clase de usuario (Browsing, Shopping y Ordering) que se menciona en TPC-W.

Se puede afirmar consecuentemente que, siguiendo el estándar propuesto por TPC-W, se ha conseguido adaptarlo mediante el generador de carga GUERNICA, a una plataforma de e-commerce actual como es Prestashop.

## 7 Conclusiones y trabajos futuros

---

Para finalizar, en este último capítulo, se exponen las conclusiones que se han obtenido al terminar el proyecto. De igual manera, se relatarán los posibles trabajos futuros que pueden derivar de este trabajo.

### 7.1. Conclusiones

Como se ha visto en los primeros capítulos, no existe hoy en día ningún estándar actualizado para benchmarks de e-commerce, aunque se tiene la especificación y la implementación de un estándar ya en desuso como es TPC-W. Además, actualmente el uso de CMS para la creación de comercios electrónicos es predominante en Internet y cada vez, es mayor el número de usuarios que realizan sus compras haciendo uso de este tipo de plataformas.

Este auge y tendencia actual de realizar cada vez más las compras a través de la red, plantea muchos desafíos, tanto para los comercios, como para los proveedores de servicios que alojan dichas plataformas de ventas. Por eso, resulta interesante el disponer de herramientas actualizadas para poder poner a prueba dichas plataformas y de esta manera, que puedan ofrecer los mejores servicios y aumentar sus ventas. Un benchmark para comercio electrónico es útil desde el punto de vista del comercio para observar el rendimiento de su página, hacer pruebas de escalado, saber el número de usuarios que soporta la página y algunas tareas más, por ejemplo. Para el proveedor de servicios, el uso de este benchmark resulta interesante para poner a prueba las infraestructuras que tengan y hagan uso de estas plataformas. De esta manera, los proveedores pueden saber, por ejemplo, si el servidor soporta x usuarios, hacer pruebas para saber si escala de forma correcta en caso de visitas masivas de usuarios en momentos clave, analizar el rendimiento a nivel Hardware de las instalaciones y un sinfín de tareas más.

En este proyecto, se propone e implementa una solución actualizada de un benchmark para e-commerce, basado en Prestashop, ajustándose a las especificaciones de TPC-W. Se ha hecho uso del generador de carga GUERNICA, con el cual se ha conseguido generar una carga válida que simula las navegaciones que propone el estándar de TPC-W sobre una tienda electrónica de libros bajo el CMS Prestashop.

De forma general, se han cumplido los objetivos del trabajo, ya que se ha llevado a cabo la revisión del estándar de TPC-W corrigiendo los problemas de implementación que este tenía y validando dichos cambios, se ha conseguido partir de unas bases de datos parejas entre TPC-W y Prestashop. Por último, se ha logrado generar las navegaciones desde GUERNICA y mediante la realización de múltiples experimentos, hemos podido validar que los resultados son

satisfactorios y correctos, realizándose las navegaciones bien y cumpliendo los porcentajes de TPC-W.

Para concluir este apartado, se ha de decir que se ha validado la adaptación y se ha implementado un entorno de pruebas funcional, pero aún faltarían cosas por hacer para poder considerarlo un benchmark completo y funcional con posibilidad de liberarlo o distribuirlo. Algunos de estas mejoras se exponen en trabajos futuros.

## **7.2. Trabajos futuros**

La adaptación se ha validado y podemos decir que lanzando carga desde GUERNICA generamos navegaciones válidas sobre la tienda de libros en Prestashop. Queda para futuros trabajos el analizar los tiempos de respuesta, el tamaño de los archivos transferidos y algunas estadísticas que podemos obtener desde GUERNICA y Apache para de esta forma, obtener resultados para saber si hemos de afinar más la plataforma.

También queda pendiente el realizar más tipos de navegaciones para poder cubrir necesidades específicas como: compradores oportunistas que buscan ofertas, clientes fieles, descuentos especiales para ciertos clientes, clientes potenciales, clientes que comparan precios, o un administrador de la web podrían ser algunas de ellas.

Otro de los posibles trabajos puede ser el adaptar este benchmark a otro CMS de e-commerce para después poder hacer experimentos y comparar ambas plataformas para saber cuál obtiene mejores resultados.

Por último, podría ser interesante hacer una integración de este entorno junto con una red social para poder ver la presencia de ventas electrónicas desde redes sociales y de que forma los usuarios responden a estos reclamos muy utilizados actualmente.

# Bibliografía

---

- [1] Página oficial de TPC-W- <http://www.tpc.org/tpcw/> (Consulta: 5 septiembre 2017)
- [2] Sistemas de Gestión de Contenidos - [https://es.wikipedia.org/wiki/Sistema\\_de\\_gesti%C3%B3n\\_de\\_contenidos](https://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_contenidos) (Consulta: 1 septiembre 2017)
- [3] Alberto Iglesias Fraga - <http://www.ticbeat.com/tecnologias/20-745-millones-de-euros-el-dinero-que-mueve-el-e-commerce-en-espana/> (Consulta: 1 septiembre 2017)
- [4] Imagen Soluciones CMS a nivel global. Recuperado de <https://www.1and1.es/digitalguide/hosting/cms/cms-en-comparativa-los-gestores-de-contenido-mas-usados/> (Consulta: 1 septiembre 2017)
- [5] Imágenes de uso de CMS a nivel mundial y en España. Recuperado de Trends Built With - <https://trends.builtwith.com/shop> (Consulta: 1 septiembre 2017)
- [6] Peña-Ortiz, Raúl & Antonio Gil, José & Sahuquillo, Julio & Pont, Ana. (2012). Providing TPC-W with web user dynamic behavior. CLEI electronic journal (CLEIej). 15. 1.
- [7] R.C Dodge JR, D.A Menasce and D. Barbara, “Testing e-commerce site scalability with TPC-W” - 2001
- [8] C. Amza, A. Chanda and A. Cox “Specification and implementation of dynamic web sites benchmarks” - 2002
- [9] D. García and J. García, “TPC-W e-commerce benchmark evaluation” – 2003
- [10] Menascé, Daniel. (2002). TPC-W: A benchmark for e-commerce. Internet Computing, IEEE. 6. 83 - 87. 10.1109/MIC.2002.1003136.
- [11] Mehdi Khouja, Farouk Kamoun, Catalina M. Lladó, Ramón Puigjaner “Experimenting with the TPC-W E-Commerce Benchmark” - Manouba 2010
- [12] Sanket K. Dhopeswarkar and Prof. Varsha Apte, “Overload Control of E-Commerce Web Servers” - M.Tech. Project Stage-2 Report”
- [13] R.C Dodge JR, D.A Menasce and D. Barbara, “Testing e-commerce site scalability with TPC-W” - 2001
- [14] Peña-Ortiz, Raúl. (2013). Accurate Workload Design For Web Performance Evaluation.

Adaptación de un Benchmark para e-commerce bajo plataformas actuales de ventas como Prestashop haciendo uso del generador de carga GUERNICA

[15] TPC BENCHMARKTM-W (Web Commerce) Specification Version 1.8 Feb 19, 2002

[16] <http://doc.prestashop.com/pages/viewpage.action?pageId=28016773>  
(Consulta: septiembre 2017)