

I.E.S. Punta del Verde

DESARROLLO DE APLICACIONES WEB

KNOWBUYSELL

Proyecto Fin de Grado

Autor:
Yolanda Morales Zamora

Junio 2024

1 Introduction

1.1 Introducción del proyecto

Este proyecto se ha elaborado a raíz de la necesidad de control de información por parte de los usuarios de una plataforma de compras, para gestionar esa información para su mejor organización económica, es decir, como usuario "quiero saber en qué gasto más dinero o menos dinero" y a su vez con la misma información obtenida generar marketing personalizado para el usuario por parte de la entidad vendedora, es decir, como empresa "quiero conocer que es lo que más compra el usuario según categoría para ofrecerle otros productos que no conozca de la misma categoría" .

Datos técnicos básicos del proyecto:

Nombre del proyecto: Knowbuysell.

Framework utilizado: Laravel, el framework de los artesanos de la web.

Lenguaje de programación: PHP, Blade, Javascript.

Lenguaje Latex para la realización de este informe.

1.2 Propósito

Desarrollar una aplicación que ayude a mejorar el control económico del usuario cuando compre a través de ella, y a la misma vez, ayude a mejorar el marketing de la entidad vendedora, a través de la información registrada. A través de la información registrada organizarla para beneficiar a la parte del "cliente" y a la parte del "servidor".

1.3 Objetivos del proyecto

Objetivos del proyecto:

- a) control por parte del usuario de la información que generan sus compras realizadas
- b) el control por parte de la entidad vendedora de la información de compra del cliente para ofrecerle más productos o servicios (mejora de marketing).

En los apartados a) y B) se reutiliza la misma información sólo que según el enfoque del cliente o servidor variará su gestión o uso.

1.4 Coste del proyecto

1.4.1 Costes de desarrollo

Los costes directos comprenden los costes de personal. Para calcular el coste de personal es necesario saber cuál es el salario bruto del trabajador y los costes sociales. Para llevar a cabo esta planificación se tiene en cuenta un contrato a tiempo completo, en el que la empresa debe abonar una cuota patronal a la seguridad social del 23,60% actualmente en España.

Estableciendo el coste medio de la hora por parte de un programador en España en 20 euros y siendo 500 las horas totales planificadas para el desarrollo del proyecto completo, se obtiene una remuneración total bruta de 10.000 euros. Tabla 1:

Concepto	Coste
Remuneración bruta	10.000 €
Costes Seguridad Social	2.360 €
TOTAL	12.360 €

Table 1: Costes directos

Para la obtención de los costes indirectos, en este caso, hay que tener en cuenta la amortización del equipo utilizado por el trabajador.

La tabla 3 muestra aquellos dispositivos hardware que están expuesto a un desgaste en el tiempo en el que se elabora el el proyecto:

1.4.2 Costes de implantación

Debemos tener en cuenta que a continuación damos estimaciones de los costes de implantación:

- En cuanto a Hosting y Servidores:
 - Hosting Compartido: 5 a 20 €/ mes
 - Servidor VPS: 20 a 100 €/ mes
 - Servidor Dedicado: 80 a 300 €/ mes
 - Servicios en la Nube (AWS, Google Cloud, Azure): 50 a 200 €/mes
 - Para un primer mes de operación:
 - Costo Inicial de Hosting: 50 a 300 €/ anual
 - Registro de Dominio: 10 a 15 €/ anual
 - Certificados SSL: 0 a 200 €/anual (Let's Encrypt gratuito o pagos de proveedores como Comodo)
Subtotal Infraestructura ((Primer Mes): 60 a 515 €
 - Licencias y Herramientas
 - IDE y Herramientas de Desarrollo: utilizamos herramientas de código abierto y gratuitas como Visual Studio Code a modo de IDE. Licencias de Software: Depende de necesidades específicas, pero generalmente bajo costo
 - Subtotal Licencias y Herramientas = 0 €
 - Seguridad y Auditorías
 - Firewall y Seguridad en la Nube (opcional): 20 a 100 €/mes
 - Auditorías de Seguridad: 500 a 1,500 €/ mes
 - Subtotal Seguridad y Auditorías: 520 a 1,600 €/ mes
- Total del importe estimado:**
- Coste fijo: 12.360 €**
 - Coste variable: 580 a 2.115 €/ mes**

2 Análisis del sistema

2.1 Introducción

En este apartado detallamos el proyecto a realizar: análisis, diseño, implementación o codificación y prueba.

2.2 Análisis de requisitos

El proyecto quiere obtener la mejora en la información económica registrada para el usuario de la aplicación así como la mejora a nivel de marketing para la empresa que presta el servicio de venta. El usuario debe poder realizar la compra de productos, descargar la información y crear un documento formato PDF, acceder a su registro, a la información de sus compras, poder comprar, modificar la compra y eliminar la compra. El usuario no podrá acceder a la información de otros usuarios, sólo a la suya propia. El administrador o empleado de la empresa propietaria de la aplicación, podrá acceder a toda la información de las compras realizadas por cada usuario, podrá modificar, eliminar cualquier registro de la base de datos, podrá obtener la información en formato gráfica de las categorías de compras realizadas por el usuario con la finalidad de mejorar el marketing para poder aumentar la compras, y por tanto los beneficios.

2.2.1 Requerimientos funcionales

Los requisitos funcionales de una aplicación web son las funciones y características específicas que la aplicación debe tener para cumplir con los objetivos y necesidades de los usuarios. En nuestro proyecto, aunque algunos no se hayan conseguido como metas serían los siguientes:

- Inicio de sesión (Conseguido)
- Registro de usuario (Conseguido)
- Gestión de perfiles (Conseguido)
- CRUD para realizar compras (Conseguido)
- Integración de terceros como un sistema de pagos (No conseguido)
- Medidas de seguridad: cifrado de claves de usuario (Conseguido)
- Doble verificación (No conseguido)

2.2.2 Requerimientos no funcionales

Los requisitos no funcionales de una aplicación web son las características que definen la calidad y el comportamiento del sistema en lugar de las funciones específicas que realiza. En nuestro caso serían los siguientes:

- Rendimiento: rapidez que se consigue con Laravel
- Disponibilidad (No conseguido)
- Seguridad: autenticación, CSRF (Conseguido)
- Usabilidad: intuitiva y claridad en el diseño (Conseguido)
- Mantenibilidad: modularización del código a través de Modelo Vista Controlador, documentación (Conseguido)
- Cumplimiento Normativo: GDPR o Legislación de Protección de datos.

2.3 Casos de uso: Diagramas y Narrativas de Caso de Uso

Los diagramas de casos de uso reflejarán las funcionalidades que correspondan a cada uno de los ámbitos concretos y más importantes del sistema. En la narrativa de los casos de uso se especifica las funciones, tareas e interacciones que se dan lugar en el sistema, es decir, se describen detalladamente cada uno de los casos de uso. Lo habitual es realizar diagramas de tipo UML, nuestro diagrama conforme al proyecto, se puede comprobar en la [1](#) como se puede constatar más abajo.

Una vez observado el diagrama [1](#), en nuestro caso, tenemos dos actores: **usuario** y **administrador**. El **usuario** gestionará las tareas indicadas en el diagrama, como son: registrar usuario, iniciar sesión, realizar compras a través de un CRUD, generar documentos en formato PDF de cada listad de compras y obtener información porcentual de las compras en gráficas para saber en qué se gasta más el dinero y cómo progresa las compras de cada usuario; por otro lado como hemos indicado, tenemos a otro actor que es el **administrador** cuyas funciones consisten en: supervisar y gestionar las posibles incidencias de las tareas o actividades generadas por del usuario, y además extraer la información almacenada del usuario y con ella generar ofertas atractivas dirigidas al usuario. Es en la tarea de obtener información porcentual en gráficas la que ambos actores tienen un punto en común, pues la actividad que genera la información porcentual del usuario, es extraída por el administrador y gracias a esta información recopilada podrá generar ofertas que posteriormente irán dirigidas al usuario, con el fin de aumentar las ventas y obtener beneficios económicos en la empresa objeto de crean de esta aplicación web, tal y como hemos indicado.

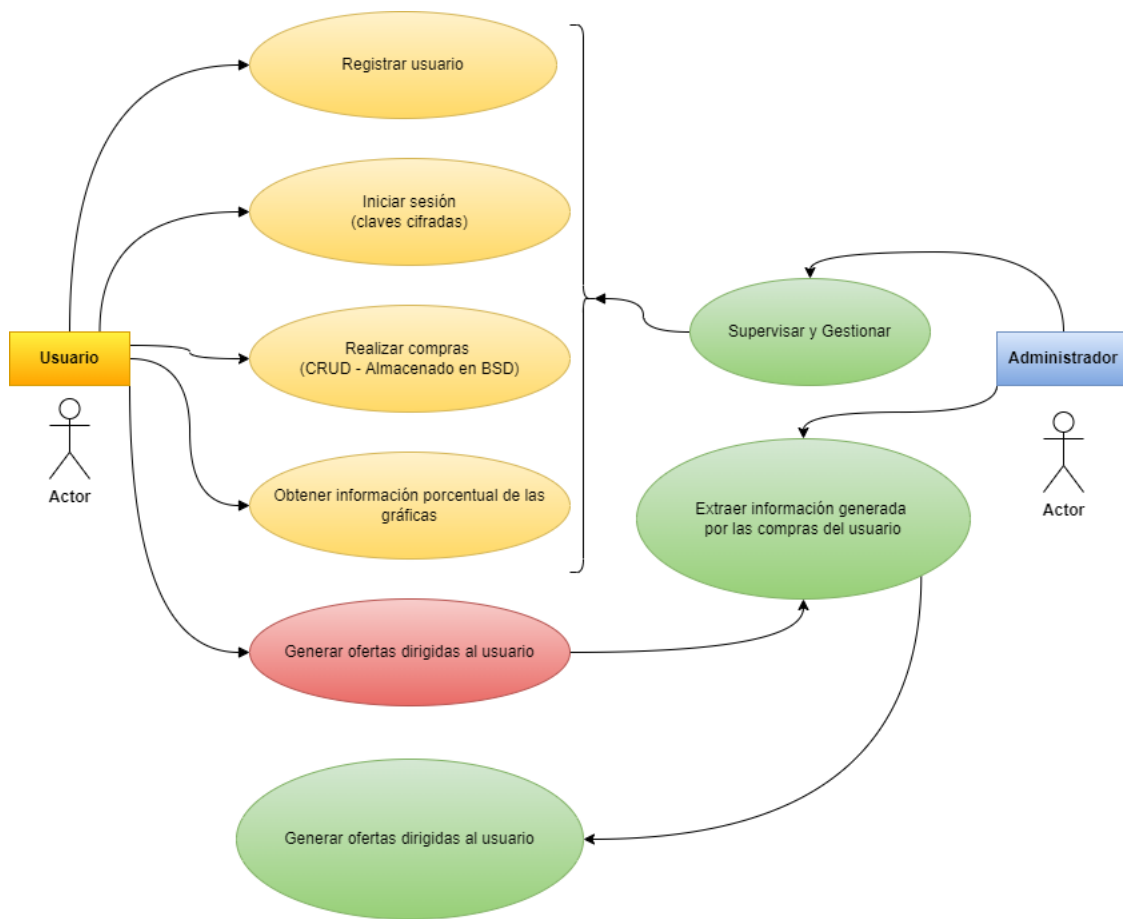


Figure 1: Diagrama de la interacción en la aplicación web: Knowbuysell.

Referencia	CU-01
Título del caso de uso	Registrar usuario
Actor	Usuario
Condiciones de entrada	El usuario no puede estar registrado previamente.
Flujo de eventos	Desde la página web de Bienvenida seleccionamos el botón "Registrar" que nos lleva a otra página donde registramos el usuario, el email y dos veces la clave para validarla. Después de ello seleccionamos el botón "Create account".
Camino alternativo	Si el usuario está registrado desde esa página tiene la opción de pulsar el botón "Login" para meter usuario y contraseña.
Condiciones de salida	El cliente queda registrado en el sistema, por tanto, almacenado en nuestra base de datos.

Table 2: CU-01 Registrar usuario

DIAGRAMA DE CASO DE USO: APLICACIÓN WEB KNOWBUYSELL (Figura 1)

NARRATIVAS DE CASOS DE USO (Tablas 2,3,4, 5, 6 y 7)

En cada tabla se puede localizar cada Caso de Uso, en el cual se describe la referencia del Caso de Uso, el título del Caso de Uso, qué actor es el que está implicado en él, las condiciones de entrada, el flujo de eventos, el cambio alternativo en el que se puede mover hacia otra sección de la aplicación web y las condiciones de salida, es decir, cuál es el out en la aplicación en esta sección.

Referencia	CU-02
Título del caso de uso	Iniciar sesión
Actor	Usuario
Condiciones de entrada	El usuario debe estar registrado previamente.
Flujo de eventos	Desde la página web de Bienvenida seleccionamos el botón "Login" que nos lleva a otra página donde incluimos el usuario y la clave previamente registrada según CU-01. Después de ello seleccionamos el botón "Submit".
Camino alternativo	Si el usuario no está registrado desde esa página tiene la opción de pulsar el botón "Regiser" para registrar los datos de usuario.
Condiciones de salida	El cliente queda logado en el sistema, por tanto, puede acceder a las página desde la que va a poder interactuar con la aplicación.

Table 3: CU-02 Iniciar sesión

Referencia	CU-03
Título del caso de uso	Realizar compras
Actor	Usuario
Condiciones de entrada	El usuario debe estar logado.
Flujo de eventos	Desde la página web al pinchar en la palabra "Comprar" nos lleva a la página donde podemos escribir cada campo, podemos crear, modificar y eliminar los registros, el usuario se guarda de forma automática. Después de ello seleccionamos el botón "Guardar Datos"
Camino alternativo	Si cambiamos de idea podemos pinchar sobre "Ir a Home" y volvemos la web de Bienvenida.
Condiciones de salida	El cliente queda logado en el sistema, por tanto, puede acceder a las página desde la que va a poder interactuar con la aplicación.

Table 4: CU-03 Realizar compras

Referencia	CU-04
Título del caso de uso	Generar documentos PDF de listado de datos
Actor	Usuario/Administrador
Condiciones de entrada	El usuario o administrador debe estar logado.
Flujo de eventos	Desde la página web al pinchar en la palabra "Compras Realizadas" nos lleva a la página donde podemos seleccionar el botón "PDF" para generar un documento de formato .pdf y guardarlo en la ubicación que seleccionemos
Camino alternativo	Si cambiamos de idea, y no queremos generar documento en formato .pdf podemos pinchar sobre "Ir a Home" y volvemos la web de Bienvenida.
Condiciones de salida	El documento de tipo .pdf queda generado y a su vez descargado de la página web.

Table 5: CU-04 Generar documento PDF de listado de datos

Referencia	CU-05
Título del caso de uso	Obtener información porcentual a través de gráficas
Actor	Usuario/Administrador
Condiciones de entrada	El usuario o administrador debe estar logado y haber realizado compras en el apartado correspondiente.
Flujo de eventos	Desde la página web al pinchar en la palabra “Graficas” nos lleva a la página donde podemos obtener la información de todas las compras a través de gráficas con sus porcentajes
Camino alternativo	Si cambiamos de idea, o hemos terminado de ver la información podemos pinchar sobre “Ir a Home” y volvemos la web de Bienvenida.
Condiciones de salida	Habremos generado la información de las compras a través de gráficas con sus correspondientes porcentajes.

Table 6: CU-05 Obtener información porcentual a través de gráficas

Referencia	CU-06
Título del caso de uso	Generar ofertas dirigidas al usuario
Actor	Administrador
Condiciones de entrada	El administrador debe estar logado y haber extraído la información de gráficas.
Flujo de eventos	Desde la página web al pinchar en la palabra “Graficas” nos lleva a la página donde podemos obtener la información de todas las compras a través de gráficas con sus porcentajes, desde allí tenemos que añadir la opción para enviar correos de publicidad al usuario
Camino alternativo	Si cambiamos de idea, pinchamos en “Ir a Home” y volvemos la web de Bienvenida.
Condiciones de salida	Habremos generado la información de las compras a través de gráficas con sus correspondientes porcentajes.

Table 7: CU-06 Generar ofertas dirigidas al usuario

3 Diseño del sistema

3.1 Introducción

Para nuestra aplicación web hemos decidido utilizar la base de datos **MaríaDB**, aunque existen informáticos que recomiendan para Laravel utilizar **MySQLWorkbench**, pero debido al tiempo que disponemos y que no hemos utilizado nunca esta última hemos decidido utilizar **MaríaDB** porque ya tenemos conocimientos sobre ella y puede ayudarnos a no perder tiempo, pero nos apuntamos como nota estudiar y conocer el uso de **MySQLWorkbench**. Todo ello a través de servidor Apache, tanto el acceso al servidor como el uso de la base de datos lo hemos conseguido gracias a la aplicación **Xampp**.

3.2 Diagrama de clases

Diagrama que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos.

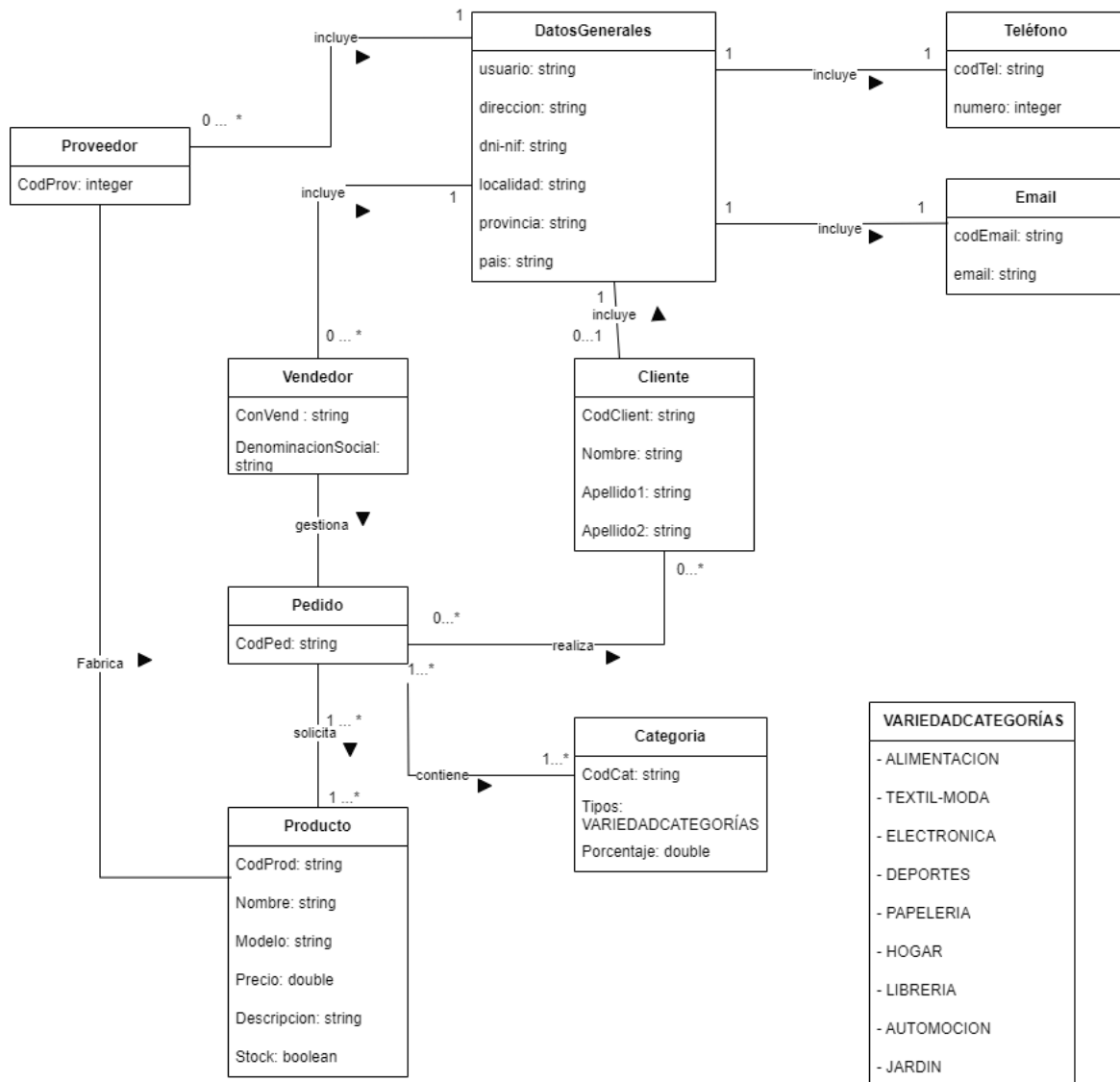


Figure 2: Diagrama de clases.

3.3 Diseño de la Base de Datos.

3.3.1 Diseño Lógico.

Nuestro diseño de bases de datos empezó siendo muy ambicioso (ver figura 3), finalmente ha quedado mucho más reducida.

Entidades, atributos, claves primarias, identificadores únicos, relaciones entre las entidades.

3.3.2 Diseño Conceptual.

Tablas, campos, tipos de datos, restricciones, claves primarias, claves foráneas, triggers de validación si los hubiera.

3.4 Diseño de la Interfaz.

En este apartado se define cual va a ser la apariencia visual de la aplicación: interfaz visual entre el usuario y la aplicación. En este apartado se debe explicar cómo son las plantillas en las que se basa la apariencia de nuestra interfaz así como cada una de las pantallas que tengamos.

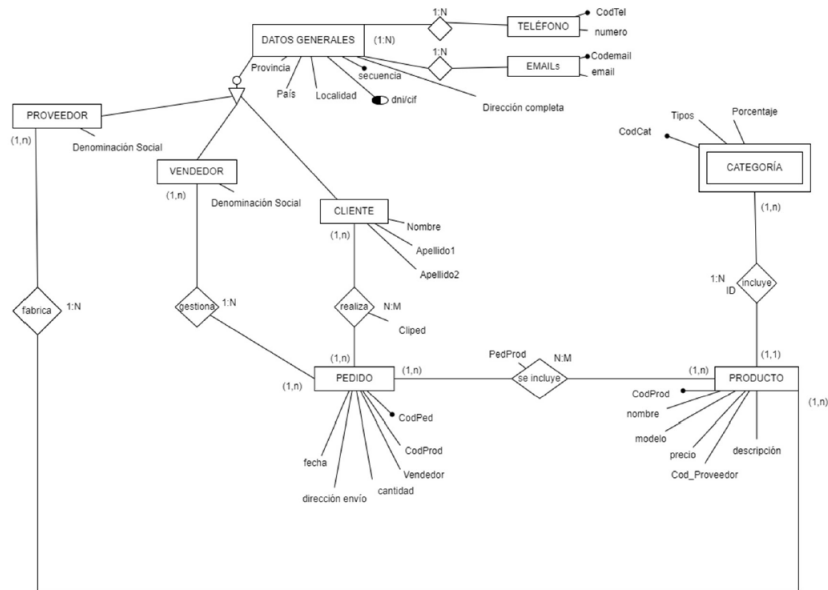


Figure 3: Diseño de Base de Datos inicial.

4 Implementación

4.1 Introducción

En este momento ya se encuentra definido el problema y la solución, por lo que lo siguiente será transformar el modelo obtenido en las actividades anteriores en código fuente.

4.2 Arquitectura Cliente/Servidor

Explicar en qué consiste

4.3 Lenguajes de Programación.

El lenguaje de programación principal utilizado en este proyecto es PHP, después hemos utilizado plantillas BLADE, también se incluye Javascript, Latex y como lenguaje de marcas HTML utilizando también CSS.

Hemos optado por PHP porque suponía un reto para el proyecto, puesto que los lenguajes con los que tenemos más preferencia de trabajo son JAVA y PHYTON.

4.4 Herramientas de Desarrollo

Explicar que herramientas has utilizado para desarrollar tu aplicación

4.5 Codificación

Todo el código y la información se encuentran ubicados en GitHub, se puede acceder a través de <https://github.com/ymorzam328/KnowBuySell.git>.

5 Conclusiones

5.1 Resumen de conclusiones

Ver en Anexo 1 a este documento, donde también se pueden encontrar las Referencias o Bibliografía utilizada.