

Block factory

Construye tu aprobado



Tecnologías avanzadas del desarrollo

Alberto BALSERA LÓPEZ, STEPHAN PEREZ MARIN, ENRIQUE MACHADO DOMINGUEZ

2023

Contenido

[Mockups 1](#_Toc135856393)

[Casos de uso 3](#_Toc135856394)

[Enlace 8](#_Toc135856395)

[Instrucciones uso 8](#_Toc135856396)

[Estado del arte 8](#_Toc135856397)

[Metodología de desarrollo. 9](#_Toc135856398)

[Base de datos. 10](#_Toc135856399)

[Interfaz. 11](#_Toc135856400)

[Implementación. 15](#_Toc135856401)

[Justificación. 16](#_Toc135856402)

[Conclusiones. 16](#_Toc135856403)

[Lecciones aprendidas. 16](#_Toc135856404)

[ENLACE: 17](#_Toc135856405)

[Instrucciones de uso principal de la página: 17](#_Toc135856406)

# Mockups

Con el fin de minimizar la complicación de la página se ha optado por reducir el número de botones y texto en pantalla al mínimo. Los botones tienen colores que contrastan con el color de fondo el cual mantiene la temática. Los colores de los botones siguen la lógica de verde o azul para agregar o confirmar y rojo para deshacer o borrar.

Gráfico

Descripción generada automáticamente

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Escala de tiempo

Descripción generada automáticamente con confianza media

# Casos de uso

Diagrama

Descripción generada automáticamente

|  |  |
| --- | --- |
| **Caso de Uso 1** | Log in |
| **Precondición** | Usuario previamente registrado |
| **Descripción** | El usuario inicia sesión en el sistema |
| **Secuencia normal** | 1. El cliente da click el login 2. El sistema pide usuario y contraseña 3. El cliente introduce 4. El sistema valida y da acceso |
| **Postcondición** | El usuario está dentro del sistema y accede a todas las funcionalidades |
| **Excepciones** | * 1. El sistema no da acceso al no encontrarse registrado |

|  |  |
| --- | --- |
| **Caso de Uso 2** | Log out |
| **Precondición** | Usuario previamente logeado |
| **Descripción** | El usuario sale del sistema |
| **Secuencia normal** | 1. El cliente da click el log out 2. El sistema sale y redirige a log in (Caso de Uso 1) |
| **Postcondición** | El usuario está fuera del sistema y no accede a ninguna de las funcionalidades |
| **Excepciones** |  |

|  |  |
| --- | --- |
| **Caso de Uso 3** | Alta usuario |
| **Precondición** |  |
| **Descripción** | El usuario se registra en el sistema |
| **Secuencia normal** | 1. El cliente da click el registrar 2. El sistema pide nombre, email y contraseña 3. El cliente introduce 4. El sistema registra y redirige a log in |
| **Postcondición** | El usuario se registra en el sistema y accede a todas las funcionalidades |
| **Excepciones** | 4.1. El sistema no registra por error al introducir los datos |

|  |  |
| --- | --- |
| **Caso de Uso 4** | Listar productos |
| **Precondición** | El cliente está logueado |
| **Descripción** | El sistema muestra todos los productos disponibles |
| **Secuencia normal** | 1. El cliente al loguearse, se encuentra en el Inicio con la lista de todos los productos |
| **Postcondición** | - |
| **Excepciones** | - |

|  |  |
| --- | --- |
| **Caso de Uso 5** | Visualizar producto |
| **Precondición** | El cliente está logueado |
| **Descripción** | El sistema muestra un producto en concreto |
| **Secuencia normal** | 1. El cliente da click en ver más de un producto específico, 2. El sistema redirige a otra pantalla con información de ese producto |
| **Postcondición** | El cliente ve más datos |
| **Excepciones** |  |

|  |  |
| --- | --- |
| **Caso de Uso 6** | Añadir producto al carrito |
| **Precondición** | El cliente está logueado |
| **Descripción** | El sistema añade un producto al carrito de compra |
| **Secuencia normal** | 1. El cliente da click en añadir al carrito 2. El sistema añade el producto al carrito |
| **Postcondición** | El cliente puede seguir viendo productos y añadir más, si quiere. |
| **Excepciones** |  |

|  |  |
| --- | --- |
| **Caso de Uso 7** | Realizar compra carrito |
| **Precondición** | El cliente está logueado |
| **Descripción** | El cliente compra lo que hay en el carrito |
| **Secuencia normal** | 1. El cliente da click en comprar, todo lo que hay en el carrito 2. El sistema accede al proceso de compra |
| **Postcondición** | Se vacía el carrito |
| **Excepciones** |  |

|  |  |
| --- | --- |
| **Caso de Uso 8** | Visualizar contenido carrito |
| **Precondición** | El cliente esta logueado y ha añadido algo al carrito |
| **Descripción** | El cliente visualiza lo que ha agregado al carrito |
| **Secuencia normal** | 1. El cliente da click en visualizar contenido del carrito 2. El sistema muestra la lista de productos del carrito. |
| **Postcondición** | Se vacía el carrito |
| **Excepciones** | 2.1.No hay nada en el carrito para mostrar |

|  |  |
| --- | --- |
| **Caso de Uso 9** | Añadir favorito |
| **Precondición** | Se está visualizando un producto |
| **Descripción** | El cliente añade un producto a sus favoritos |
| **Secuencia normal** | 1. El Usuario presiona el botón de la estrella junto al producto |
| **Postcondición** | Se añade el producto a favoritos y se puede observar el cambio ene l boton |
| **Excepciones** |  |

|  |  |
| --- | --- |
| **Caso de Uso 10** | Eliminar favorito |
| **Precondición** | El cliente ha añadido el producto previamente a favoritos |
| **Descripción** | El cliente elimina un producto de sus favoritos |
| **Secuencia normal** | 1. El Usuario presiona el botón de la estrella junto al producto |
| **Postcondición** | Se vacía el carrito |
| **Excepciones** | 2.1.No hay nada en el carrito para mostrar |

|  |  |
| --- | --- |
| **Caso de Uso 11** | Filtrar favoritos |
| **Precondición** | El cliente esta logueado y está visualizando los productos |
| **Descripción** | El cliente visualiza los productos que previamente ha agregado a favoritos |
| **Secuencia normal** | 1. El cliente presiona el botón para filtrar sus favoritos |
| **Postcondición** | Se visualizan solo los productos que están en favoritos |
| **Excepciones** | 1.1.No hay nada para mostrar |

|  |  |
| --- | --- |
| **Caso de Uso 12** | Listar categorías |
| **Precondición** | El Administrador esta logueado |
| **Descripción** | El administrador visualiza las categorías existentes |
| **Secuencia normal** | 1. El administrador selecciona el botón del menú superior “Categorías” |
| **Postcondición** | Se visualizan las categorías |
| **Excepciones** | 2.1.No hay categorías |

|  |  |
| --- | --- |
| **Caso de Uso 13** | Crear categorías |
| **Precondición** | El cliente esta logueado y ha añadido algo al carrito |
| **Descripción** | El cliente visualiza lo que ha agregado al carrito |
| **Secuencia normal** | 1. El administrador da click en crear nueva categoría 2. El sistema muestra el formulario de categoría 3. El usuario completa el formulario y lo envía |
| **Postcondición** | Se crea una nueva categoría |
| **Excepciones** |  |

|  |  |
| --- | --- |
| **Caso de Uso 14** | actualizar categorías |
| **Precondición** | El administrador esta visualizando las categorías |
| **Descripción** | El administrador actualiza una categoría |
| **Secuencia normal** | 1. El administrador selecciona el botón de editar una categoría |
| **Postcondición** | Se modifica una categoría |
| **Excepciones** |  |

|  |  |
| --- | --- |
| **Caso de Uso 15** | Eliminar categorías |
| **Precondición** | El administrador está visualizando las categorías |
| **Descripción** | El administrador elimina una de las categorías |
| **Secuencia normal** | 1. El administrador selecciona el botón de eliminar una categoría |
| **Postcondición** | Se elimina la categoría |
| **Excepciones** |  |

# Enlace

<https://github.com/AruberutoEXE/BlockFactory>

# Instrucciones uso

Clonar el repositorio de git.

Agregar las dependencias con composer.

Ejecutar las migrates (o el .sql).

Ejecutar los seeders.

Desplegar la página con serve.

Acceder a la ruta principal de la página (<http://127.0.0.1:8000>).

En ella podremos observar los productos del catálogo.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Podemos añadirlos al carrito o ver más detalles.

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Una vez agregados al carrito podemos pulsar el apartado de carrito y visualizar nuestra compra.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Si todo es correcto podemos finalizar la compra.

# Estado del arte

El estado del arte del comercio electrónico está en constante evolución debido al rápido avance de la tecnología y las cambiantes preferencias de los consumidores.

* Laravel

Laravel es un framework de desarrollo web PHP de código abierto que sigue el patrón de diseño Modelo-Vista-Controlador (MVC). Proporciona una sintaxis elegante y fácil de usar, así como una amplia gama de características y funcionalidades para agilizar el desarrollo de aplicaciones web. Laravel ofrece una arquitectura modular, enrutamiento flexible, migraciones de bases de datos, autenticación de usuarios, entre muchas otras características útiles.

* MySql

MySQL es un sistema de gestión de bases de datos relacional ampliamente utilizado y de código abierto. Es conocido por su rendimiento, escalabilidad y confiabilidad. MySQL proporciona una amplia gama de características para administrar y manipular datos, incluyendo consultas SQL, índices, integridad referencial, transacciones y funciones avanzadas de almacenamiento y búsqueda.

* Xampp

XAMPP, por su parte, es un paquete de software gratuito y multiplataforma que proporciona un entorno de desarrollo local para aplicaciones web. Incluye componentes como el servidor web Apache, el sistema de gestión de bases de datos MySQL, el intérprete de scripts PHP y el servidor de correo electrónico Mercury. XAMPP simplifica la configuración y el inicio de un entorno de desarrollo web completo en tu propio ordenador, lo que te permite desarrollar y probar tu aplicación de forma local antes de desplegarla en un entorno de producción.

# Metodología de desarrollo.

Se ha usado Git y GitHub para una colaboración eficiente con un seguimiento de cambios precisos y sus correspondientes aprobaciones. El código está mejor gestionado y facilita la coordinación del equipo para llegar al objetivo.

Todo ello con un repositorio, ramas,  sincronización de cambios y resolución de conflictos

La metodología está basada en la asignación secuencial de tareas a cada miembro del equipo, donde cada persona se enfoca en completar un número específico de tareas antes de pasar a la siguiente. Está metodología se gestiona en las siguientes etapas.

1. Definición de tareas: En primer lugar, se debe realizar una identificación y definición clara de todas las tareas que deben completarse en el proyecto. Estas tareas deben ser específicas, medibles y alcanzables. Pueden dividirse en subtareas más pequeñas si es necesario para una mejor organización.

2. Asignación de tareas: Una vez que las tareas estén definidas, se asignará a cada miembro del equipo un número específico de tareas para que se encargue de ellas. Es importante considerar las habilidades y fortalezas individuales al asignar las tareas, de modo que cada miembro pueda trabajar en áreas en las que sea más eficiente y productivo.

3. Enfoque en una tarea a la vez: Cada miembro del equipo se centrará en completar una tarea a la vez. Esto implica dedicar tiempo y esfuerzo exclusivamente a esa tarea particular hasta que se haya terminado o se haya alcanzado un punto de referencia significativo.

4. Revisión y colaboración: A medida que cada tarea se completa, se llevará a cabo una revisión interna o una colaboración con otros miembros del equipo, según sea necesario. Esto puede implicar la revisión del trabajo realizado, compartir conocimientos y solucionar problemas en conjunto.

5. Avance hacia la siguiente tarea: Después de que se complete una tarea y se haya obtenido la aprobación o validación necesaria, el miembro del equipo pasará a la siguiente tarea asignada en su lista. Se repite el proceso, centrándose en una tarea a la vez hasta que se cumpla con todas las tareas asignadas.

6. Comunicación y coordinación: Durante todo el proceso, es esencial mantener una comunicación clara y constante entre los miembros del equipo. Esto puede incluir actualizaciones regulares sobre el progreso, compartir información relevante, solicitar ayuda si es necesario y coordinar los esfuerzos para garantizar una colaboración efectiva.

7. Seguimiento y ajuste: Durante el desarrollo del proyecto, es importante realizar un seguimiento regular del avance general y realizar ajustes si es necesario. Esto implica evaluar el rendimiento, identificar posibles retrasos o cuellos de botella, y tomar medidas correctivas para mantener el flujo de trabajo y cumplir con los plazos establecidos.

Esta metodología de trabajo basada en asignar un número de tareas a cada miembro del equipo y enfocarse en una tarea a la vez puede ayudar a mantener la concentración y la eficiencia, al tiempo que permite una división clara de responsabilidades. Sin embargo, es importante tener en cuenta que la comunicación, la colaboración y la flexibilidad siguen siendo elementos fundamentales para garantizar un trabajo en equipo efectivo y exitoso.

# Base de datos.

Diagrama

Descripción generada automáticamente

# Interfaz.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Registro.

Inicio sesión.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Catálogo productos.

Producto detallado.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Desplegable cambiar idioma.

Categorías.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteInterfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Desplegable perfil.

Actualizar datos perfil.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Carro de compra.

# Implementación.

* Autenticación usuario: sistema seguro para poder registrarse o iniciar sesión y comprar tranquilamente. Con framework Fortify.
* Middleware: conecta partes de nuestro proyecto. Ejemplo: para el cambio de idioma en la web se usa un Middleware
* Seguridad: se evita la inyección de código SQL, por ejemplo, en el login o registro.

# Justificación.

Satisfacción del cliente: es posible poder comprar un producto correctamente.

Experiencia positiva del usuario navegando en la web: experiencia fluida con notificaciones tras realizar acciones.

Cumplimiento objetivos proyecto: a tiempo y sin sobrepasar el coste.

Patrón de fondo

Descripción generada automáticamente

# Conclusiones.

Se ha aprendido un framework de última generación, como es Laravel. Además, se ha fomentado el trabajo en equipo acorde a una planificación de entregas que cumplir. Con Git y GitHub se ha seguido la rutina de los proyectos en las empresas. Se ha integrado un paquete externo de autenticación como Fortify. Capacidad de resolución de problemas en el código y en la BBDD.

# Lecciones aprendidas.

Al principio del proyecto, se debería de haber entendido mejor el paradigma de programación de Laravel. Hacer una documentación más completa del proyecto y haber entendido mejor los requisitos que se pedían

Se ha seguido, casi en todo momento, una buena planificación. Se han hecho pruebas en todo momento para comprobar que las funcionalidades básicas funcionaban. Buena comunicación entre todos los compañeros del grupo.