

PROYECTO FINAL DOCUMENTACIÓN

Sergio Del Pozo Checa

INDICE

I. Introducción

- a. Presentación del proyecto
- b. Objetivos del proyecto
- c. Justificación del proyecto

II. Análisis de requerimientos

- a. Identificación de necesidades y requerimientos
- b. Identificación de público
- c. Estudio de mercado y competencia

III. Diseño y planificación

- a. Definición de la arquitectura del proyecto
- b. Diseño de la interfaz de usuario
- c. Planificación de las tareas y los recursos necesarios

IV. Implementación y pruebas

- a. Desarrollo de las funcionalidades del proyecto
- b. Pruebas unitarias y de integración
- c. Corrección de errores y optimización del rendimiento

V. Documentación a. Documentación técnica

- b. Documentación de usuario
- c. Manual de instalación y configuración

VI. Mantenimiento y evolución

- a. Plan de mantenimiento y soporte
- b. Identificación de posibles mejoras y evolución del proyecto
- c. Actualizaciones y mejoras futuras

VII. Conclusiones

- a. Evaluación del proyecto
- b. Cumplimiento de objetivos y requisitos
- c. Lecciones aprendidas y recomendaciones para futuros proyectos

VIII. Bibliografía y referencias

- a. Fuentes utilizadas en el proyecto
- b. Referencias y enlaces de interés

I. Introducción

a. Presentación del proyecto

Os presento mi proyecto final de grado de una página web de venta de videojuegos, el cual creo que es muy funcional y una plataforma interesante para los entusiastas de los videojuegos.

La página web, denominada “**GameLoot**”, está diseñada para ofrecer una experiencia de compra de videojuegos fácil e intuitiva para los usuarios. La plataforma cuenta con una amplia gama de videojuegos para diferentes plataformas, como PC, consolas, etc, cubriendo todas las principales categorías y géneros.

A continuación, me gustaría destacar algunas características clave de **GameLoot**:

1. Diseño y experiencia del usuario: La página web cuenta con un diseño minimalista, sencillo, elegante, con una interfaz intuitiva y fácil de navegar. Se da especial atención a la usabilidad, asegurando que los usuarios puedan interactuar con la página para poder crear tu propia lista de deseos y encontrar rápidamente los juegos que deseen comprar y acceder a información sobre ellos.
2. Catálogo amplio y actualizado: **GameLoot** ofrece un amplio catálogo de juegos, que incluirá lanzamientos recientes, clásicos populares y títulos independientes.
3. Funcionalidad de búsqueda: La página web contiene un motor de búsqueda que permitirá a los usuarios buscar juegos por el título.
4. Seguridad: La página garantiza la seguridad de los datos de los usuarios y ofrece la compra de videojuegos segura y fiable.

b. Objetivos del proyecto

Este proyecto de página web de venta de videojuegos, **GameLoot**, tiene como objetivo:

1. Crear una plataforma de venta de videojuegos en línea que ofrezca la experiencia de comprar fácil y intuitiva para los usuarios.
2. Establecer un catálogo amplio y actualizado de juegos para diferentes plataformas (PC, consolas, etc) que abarque una variedad de géneros y categorías.
3. Proporcionar un diseño minimalista y elegante en la página web, con una interfaz intuitiva y fácil de navegar.
4. Garantizar la seguridad de los datos de los usuarios.
5. Monitorear y analizar contentamente el rendimiento de la plataforma, obteniendo las opiniones de los usuarios y realizando mejoras continuas para optimizar la experiencia de los clientes.
6. Potenciar el crecimiento y expansión del mercado de los videojuegos, que es una continua evolución con el lanzamiento de nuevos títulos.

c. Justificación del proyecto

El proyecto **GameLoot** surge como respuesta a la creciente demanda y popularidad de los videojuegos, así como a la creciente tendencia de compra de juegos en línea.

La industria de los videojuegos es una de las más dinámicas a nivel mundial. El mercado de los videojuegos continúa creciendo rápidamente, con un número cada vez mayor de jugadores y un aumento en el gasto en videojuegos.

Los consumidores están optando cada vez más por realizar compras en línea debido a su conveniencia y accesibilidad. La compra de videojuegos en línea se ha convertido en la opción preferida para muchos jugadores, ya que permite acceder a una amplia gama de títulos sin tener que desplazarse a tiendas físicas. El desarrollo de una plataforma de venta de videojuegos en línea como **GameLoot** permite ofrecer una experiencia de compra mejorada en comparación en las tiendas físicas.

En resumen, el proyecto **GameLoot** se justifica debido al crecimiento continuo de la industria de los videojuegos, la preferencia de los consumidores por las compras en línea, la necesidad de una experiencia de compra mejorada, la seguridad y confianza que se brinda y el crecimiento y expansión en el mercado. Estas justificaciones respaldan la viabilidad y el éxito potencial de este proyecto.

II. Análisis de requerimientos

a. Identificación de necesidades y requerimientos

Durante la etapa de identificación de necesidades y requerimientos del proyecto **GameLoot**, he identificado los siguientes aspectos clave:

1. Catálogo amplio y actualizado: Los usuarios necesitan tener acceso a un catálogo diverso de videojuegos que abarque diferentes plataformas y géneros.
2. Experiencia de usuarios intuitiva: Los usuarios esperan una interfaz de usuarios intuitivas y fácil de navegar. Debe ser sencillo encontrar y explorar juegos, acceder a información detallada sobre ellos.
3. Funciones de búsqueda: Los usuarios necesitan opciones de búsqueda para encontrar juegos específicos por títulos. La plataforma permite a los usuarios refinar sus búsquedas y descubrir nuevos juegos basados en sus preferencias.
4. Seguridad de compras y datos: Los usuarios necesitan tener la confianza de que sus datos personales y de pago estarán protegidos de manera segura y efectiva. La plataforma implementa medidas de seguridad robustas.
5. Adaptabilidad y escalabilidad: La plataforma es adaptable para acomodar el crecimiento del catálogo de juegos, el aumento en el número de usuarios y las futuras actualizaciones y mejoras del sistema.

Al identificar estas necesidades y requerimientos, se garantiza el que el proyecto **GameLoot** se ha desarrollado teniendo en cuenta las expectativas y prioridades de los usuarios, lo que resaltarán en una experiencia de compra de videojuegos satisfactoria.

b. Identificación de público

El público principal de **GameLoot** son los jugadores entusiastas de videojuegos de todas las edades. Estos jugadores tienen un interés y pasión por los videojuegos y están dispuestos a invertir tiempo y dinero en su afición. Son personas que disfrutan de una amplia variedad de géneros y plataformas de juegos.

Además de los jugadores entusiastas, **GameLoot** también busca atraer a los gamers ocasionales. Estos son jugadores que no dedican tanto tiempo a los videojuegos, pero aún así disfrutan de la experiencia de juego y pueden estar interesados en descubrir nuevos títulos o jugar ocasionalmente.

El objetivo principal es abarcar a una amplia gama de edades. Desde adolescentes hasta adultos jóvenes y adultos de mediana edad, la plataforma busca proporcionar una oferta de juegos diversas y atractivas que se adapten a diferentes preferencias.

También se dirige a jugadores que utilizan diferentes plataformas para jugar, como PC o consola. La plataforma presenta opciones para satisfacer las preferencias de cada tipo de jugador.

El público de **GameLoot** es aquel que se siente cómodo con las compras en línea y la experiencia de navegación en plataformas digitales. Son personas que buscan comodidad, acceso rápido y una experiencia de compra optimizada en el entorno en línea.

c. Estudio de mercado y competencia

Estudio de mercado:

El mercado de los videojuegos es uno de los más grandes y de más rápido crecimiento en la industria del entretenimiento. Se estima que su valor supera los miles de millones de dólares y continúa expandiéndose a medida que más personas se interesan por los videojuegos. Existe una tendencia creciente hacia las compras en línea de videojuegos, ya que los consumidores buscan la comodidad y accesibilidad de la compra digital. Los videojuegos se juegan en una variedad de plataformas, incluyendo PC y consolas. Los jugadores tienen preferencias individuales en cuanto a la plataforma que utilizan, lo que requiere una amplia oferta de juegos para satisfacer las necesidades de cada plataforma.

Los jugadores buscan una amplia variedad de géneros, estilos y temáticas de videojuegos. Existe una demanda continua de nuevos lanzamientos, así como de juegos clásicos y populares que ofrecen experiencias únicas.

Evaluación de la competencia:

Existen varias plataformas establecidas en el mercado que ofrecen la venta de videojuegos en línea, como Steam, Eneba, Epic Games Store, Game, Media Mark, PlayStation Store. Estas plataformas tienen una amplia base de usuarios y un catálogo extenso de juegos.

Aunque la venta de videojuegos en línea ha aumentado, las distribuciones físicas aún tienen presencia en el mercado. Tiendas minoristas especializadas y grandes cadenas ofrecen videojuegos físicos, lo que puede ser una competencia directa en términos de disponibilidad y conveniencia.

Además de las plataformas de venta de videojuegos en línea, existen otros servicios y plataformas relacionadas que puedan representar competencia indirecta. Esto incluye servicios de streaming de videojuegos, suscripciones de juegos y comunidades en línea donde los juegos puedan compartir y descubrir juegos.

Para destacarse en este mercado competitivo, **GameLoot** ofrecerá propuestas de valor únicas, como un catálogo diverso y actualizado, una experiencia de usuario intuitiva, funciones sociales y personalización, seguridad y confianza en las transacciones, así como una estrategia de marketing efectiva para atraer y retener a los usuarios.

III. Diseño y planificación

a. Definición de la arquitectura del proyecto

Arquitectura del proyecto:

1. Front-end: Esta es la parte visible y accesible para los usuarios. Se ha desarrollado un front-end atractivo y de fácil navegación utilizando tecnologías como HTML, CSS, JavaScript y JQuery. La interfaz del usuario permitirá a los usuarios explorar el catálogo de juegos, buscar, realizar compras de manera intuitiva y poder crear una lista de deseos.
2. Back-end y base de datos: El back-end se encargará de gestionar la lógica de la página web y la interacción con la base de datos. Se han utilizado tecnologías como PHP para el desarrollo del back-end. La base de datos que en este proyecto he utilizado es una base de datos MySQL que se encargará de almacenar información sobre los juegos, usuarios, productos, pedidos y otros datos relevantes.
3. Motor de búsqueda: Se ha desarrollado un motor de búsqueda que permite a los usuarios buscar juegos por el título.
4. Seguridad: Se ha implementado medidas de seguridad robustas para proteger los datos de los usuarios. Se utilizan protocolos de cifrado y métodos de autenticación seguros para garantizar la confidencialidad y la integridad de la información.

Esta arquitectura para el proyecto **GameLoot** puede adaptarse y ajustarse según las necesidades específicas y los recursos disponibles. La implementación efectiva de esta arquitectura asegura un funcionamiento sólido y una experiencia de usuario satisfactoria en la plataforma de venta de videojuegos en línea.

b. Diseño de la interfaz de usuario

El diseño de la interfaz de los usuarios es minimalista, atractivo y debe captar la atención de los usuarios. Utiliza una combinación de colores sencillos y agradables a la vista, junto con un diseño limpio y organizado. El uso de imágenes relacionados con los videojuegos puede ayudar a crear un ambiente temático.

La interfaz es fácil de navegar y permite a los usuarios encontrar rápidamente los juegos que están buscando. Utiliza una estructura de menú clara y categorías bien definidas para facilitar la exploración. Incorpora un buscador visible y funcional para una búsqueda más precisa. Cada juego tiene una página individual con información completa y detallada. Incluye imágenes del juego, descripción y precio.

Implementa un carrito de compras donde los usuarios puedan agregar juegos y realizar un proceso de pago sencillo y seguro.

La página web **GameLoot** tiene un diseño responsive, es decir, que se adapta a diferentes tamaños de pantalla y dispositivos. Esto garantiza una experiencia óptima para los usuarios tanto en computadores de escritorio como en dispositivos móviles.

El diseño de la interfaz del usuario está centrada en la facilidad de uso, la estética atractiva y la experiencia del usuario.

c. Planificación de las tareas y los recursos necesarios

La página web **GameLoot** se ha planificado de la siguiente manera:

1. Pensar que tecnologías iba a utilizar, entornos de desarrollo y lenguajes de programación.
2. Creé la base de datos en phpMyAdmin y decidí usar MySQL.
3. Hice las tablas y todos los campos que necesitaba para poder realizar la página.
4. Cuando ya tenía la base de datos completa y relacionada realicé todo el sistema de carpetas del proyecto para tenerlo todo organizado.
5. Después hice el mapeo y comencé con la creación de los controladores, las vistas, etc.
6. Empecé creando la vista y la funcionalidad del login y registrar para que los usuarios tuvieran sus cuentas y sus contraseñas cifradas por temas de seguridad
7. A continuación creé la página de inicio la cual iba a mostrar todos los productos que tenía guardado en la base de datos concretamente en la tabla productos.
8. Una vez terminado la página de inicio me puse creando el modo administrador que solo tienen acceso los usuarios con rol administrado.
9. Le puse funcionalidad como insertar productos en la base de datos, borrarlos y también poder modificarlos.
10. Terminando con la funcionalidad también le añadí una opción para poder ver todos los usuarios que se habían creado y dentro la oportunidad de poder observar todos los pedidos que había realizado cada usuario, todo esto lo hice para que el administrador tuviera control total y poder ver todo lo que va ocurriendo en la página.
11. Luego comencé con las funciones y botones de favoritos para crear la lista de deseos de cada usuario y el carrito para los pedidos que vayan a realizar cada usuario, todo esto para que el usuario pueda interactuar con la página.
12. Después de realizar las dos funciones de favorito y carrito que las hice simultáneamente porque son muy parecidas, realicé las facturas. Cuando los usuarios realizasen los pedidos pudieran descargarse la factura que es un pdf con todos los detalles del pedido realizado por el usuario.
13. A continuación puse una paginación en la página de inicio donde salen 12 productos por cada página y es dinámico cuantos más producto inserte el administrador mas páginas se van a ir creando.
14. Una vez terminada la paginación hice un buscador para que el usuario pudiera encontrar los productos mas rápido.
15. Y por último hice abajo de la página un footer con los datos de **GameLoot** y un apartado sobre mí donde muestro una carta de presentación y mi currículum. Y lo último que me falta por hacer era darle estilos a todas las vistas que tiene la página y que sea responsive para cualquier dispositivo.

Los recursos que he necesitado son los siguientes:

1. Para poder realizar la página como anteriormente he comentado creé las base de datos desde phpMyAdmin y las tablas y los campos.
2. A la hora de crear el login y registrar pues lo realicé con PHP nativo utilizando formularios, mandándolos al controlador, recogiendo y haciendo las comprobaciones necesarias para que sea todo seguro y no hubiera ningún error como por ejemplo, cifrando las contraseñas, comprobando si el email existe o no existe,etc.
3. La página principal y todas las vistas que tiene **GameLoot** están realizadas con PHP nativo que es yendo al controlador y mandando todos los datos de las tablas que necesita cada vista, como por ejemplo, en la página de inicio va al controlador donde tengo un método que llamo que realiza una sql para coger todos los productos de la base de datos y ya en la vista una vez los recoge los recorro y muestro los datos de todos los productos.
4. En las vistas de ver todos los productos en el carrito y en favoritos, utilizo una sql donde uso el JOIN para poder mostrar los datos solo de los productos que se encuentran en la tabla favoritos y carrito.
5. La opción de búsqueda también utilizo una sql donde uso el LIKE, donde le paso el texto que introduce el usuario en el input y se lo meto a la sql como condición y ya solo muestra los productos que contengan ese texto, por ejemplo, si el usuario introduce una H le va aparecer todos los productos cuyo nombre empiecen o contenga la letra H.

- Después lo mas interesante son las funciones de añadir al carrito y a favoritos donde la vista tiene un script donde realiza una petición AJAX, utilizando JQuery, al controlador donde llamaba a un método en el controlador donde ya añadía a la tabla favoritos o carrito el producto que haya seleccionado el usuario.
- Para realizar las facturas me instale una librería llamada DOMPdf donde te convertía todo el código html en un pdf y solo tuve que realizar el html y darle estilos y esa librería te lo muestra en un pdf y lo puedes descargar, imprimir, etc.
- Para los estilos de las vistas he utilizado CSS y en algunas vistas Bootstrap.

IV. Implementación y pruebas

a. Desarrollo de las funcionalidades del proyecto

GameLoot implementa diferentes características y acciones que los usuarios podrán realizar. A continuación, explicaré algunas de las funcionalidades:

- Registro de usuarios: Permite a los usuarios crear una cuenta en **GameLoot** proporcionando información básica, como nombre, dirección de correo electrónico y contraseña.
- Inicio de sesión y autenticación: Permite a los usuarios iniciar sesión en **GameLoot** utilizando sus credenciales.
- Catálogo de videojuegos: Muestra una lista de videojuegos disponibles para la venta.
- Detalles del videojuego: Al hacer clic a la imagen de un videojuego, muestra información detallada, como descripción, precio e imagen.
- Carrito de compras: Permite al usuario agregar videojuegos al carrito de compras y gestionar las cantidades.
- Proceso de pedido: Permite a los usuarios completar la compra de los videojuegos en su carrito para poder realizar el pedido, y genera la factura del pedido.
- Lista de deseos: Permite a los usuarios crear su lista de deseos para guardar los videojuegos que les gusten.
- Modo Administrador: Permite solo al usuario administrador poder insertar, modificar y borrar los productos del catálogo y permite ver un listado de todos los usuarios y todos los pedidos de cada usuario.
- Buscador: Permite a los usuarios buscar videojuegos concretos para tardar menos a la hora de encontrar el videojuego que busca el usuario.
- Información sobre mí: Permite a los usuarios ver información sobre mí y observar mi currículum.

b. Pruebas unitarias y de integración

Para poder hacer las pruebas unitarias es necesario utilizar PHPUnit, es un framework de pruebas unitarias para PHP. Proporciona un conjunto de herramientas y funciones que facilitan la escritura, ejecución y verificación de pruebas automatizadas para código PHP. Explico como serían las pruebas unitarias de dos métodos de **GameLoot** bastantes importantes para su funcionalidad.

Pruebas Unitarias método **insertar_producto_favorito()**:

Este método es llamado en el controlador, lo que va hacer que cualquier usuario que quiera añadir un videojuego a su lista de deseos debe darle al icono del corazón en la página y ya entraría al controlador y se ejecutaría este método.

```
function insertar_producto_favorito(){
    $productoId = filter_var($_POST['productId'], FILTER_SANITIZE_NUMBER_INT);
    $idUser = $_SESSION['idUser'];
    $usuario = new Usuario();
    $usuario->setIdUsuario($_SESSION['idUser']);

    $productoDAO=new ProductosDAO(ConexionBD::conectar());
    $producto= new Producto();
    $producto= $productoDAO->obtenerProductoPorId($productoId);

    if($productoDAO->existeEnFavoritos($idUser, $productoId) > 0){
        //mensaje de que ya existe el producto en favoritos
    }else{
        $productoDAO->insertar_favorito($producto, $usuario);
    }
}
```

```
use PHPUnit\Framework\TestCase;
```

```
class GameLootTest extends TestCase {
```

```
    public function testInsertarProductoFavorito() {
```

```
        Preparando los datos de prueba
```

```
        $_POST['productId'] = 123; Id del producto de prueba
```

```
        $_SESSION['idUsuario'] = 456; Id del usuario de prueba
```

```
        Objetos de Mock
```

```
        $productoDAO = $this->getMockBuilder(ProductosDAO::class)
```

```
        ->disableOriginalConstructor()
```

```
        ->getMock();
```

```
        $producto = new Producto();
```

```
        Configuración de Mock de la clase ProductoDAO
```

```
        $productoDAO->expects($this->once())
```

```
        → method('existeEnFavoritos') Método que utilizo para comprobar si ya existe en la tabla favoritos o no
```

```
        ->with(456, 123) Verificar que se llama con los valores correctos
```

```
        ->willReturn(0); Producto no existe en favoritos
```

```
        Crear una instancia del objeto bajo prueba
```

```
        $objeto = new MiClase();
```

```
        Configurar la instancia del objeto bajo prueba con los mocks necesarios
```

```
        $objeto->setProductoDAO($productoDAO);
```

```
        Ejecución del método bajo prueba
```

```
        $objeto->insertar_producto_favorito();
```

```
        $productoDAO->expects($this->once())
```

```
        ->method('insertar_favorito')
```

```
        ->with($producto, $usuario); Verificar que llama con los objetos correctos como argumentos
```

```
    }
```

```
}
```

Pruebas Unitarias método **borrar_producto_favorito()**:

Este método es llamado en el controlador, cuando un usuario se encuentre en su lista de deseos y le de al icono del corazón roto borrará el producto de la lista de deseos del usuario.

```
function borrar_producto_favorito(){
    $idProducto = filter_var($_GET['id'], FILTER_SANITIZE_NUMBER_INT);
    $idusuario = $_SESSION['idUsuario'];

    $productoDAO = new ProductosDAO(ConexionBD::conectar());
    $product = $productoDAO->borrar_favorito($idProducto, $idusuario );
    if($product == true){
        header("Location: index.php?action=ir_favoritos");
    }else{
        header("Location: index.php");
        die();
    }
}
```

```
use PHPUnit\Framework\TestCase;
```

```
class MiClaseTest extends TestCase {
```

```
    public function testBorrarProductoFavorito() {
```

```
        Preparando los datos de prueba
```

```
        $_GET['id'] = 123; Id del producto de prueba
```

```
        $_SESSION['idUser'] = 456; Id del usuario de prueba
```

```
        Objetos de Mock
```

```
        $productoDAO = $this->getMockBuilder(ProductosDAO::class)
```

```
            ->disableOriginalConstructor()
```

```
            ->getMock();
```

```
        Configuración de Mock de la clase ProductoDAO
```

```
        $productoDAO->expects($this->once())
```

```
            ->method('borrar_favorito')
```

```
            ->with(123, 456)
```

```
            ->willReturn(true); Simular que el producto se borró correctamente
```

```
        Crear una instancia del objeto bajo prueba
```

```
        $objeto = new MiClase();
```

```
        Configurar la instancia del objeto bajo prueba con el mock de ProductoDAO
```

```
        $objeto->setProductoDAO($productoDAO);
```

```
        Ejecutar el método bajo prueba
```

```
        $objeto->borrar_producto_favorito();
```

```
        Verificar de resultados
```

```
        $this->assertStringContainsString("Location: index.php?action=ir_favoritos", implode(" ", headers_list()));
```

```
        Verificar que se redirija a la página correcta
```

```
    }
```

```
}
```


Para que el apartado de pruebas sea más completo voy a realizar otras dos pruebas con otros dos métodos para ver como funciona y lo que haría los métodos:

-Método para modificar productos de la base de datos (modo administrador):

El método es así

```
function modificar_producto(){
    //Recoger datos y filtrarlos
    $id = filter_var($_GET['id'], FILTER_SANITIZE_NUMBER_INT);
    $productoDAO=new ProductosDAO(ConexionBD::conectar());
    $producto= new Producto();
    $producto= $productoDAO->obtenerProductoPorId($id);

    if ($_SERVER['REQUEST_METHOD'] == 'POST') {
        //Recoger datos y filtrarlos
        $nombre=htmlentities($_POST['nombre']);
        $descripcion = htmlentities($_POST['descripcion']);
        $categoria=htmlentities($_POST['categoria']);
        $precio = htmlentities($_POST['precio']);

        //Comprobaciones para la subida de la imagen
        if (!isset($_FILES['img']) || $_FILES['img']['error'] !== UPLOAD_ERR_OK) {
            MensajeFlash::guardarMensaje('Error al cargar la imagen');
            header("Location: index.php?action=modificar_producto&id=$id");
            die();
        }

        //Comprobación del tipo de imagen que introduce el usuario
        $img_type = exif_imagetype($_FILES['img']['tmp_name']);
        if (!$img_type || !in_array($img_type, [IMAGETYPE_JPEG, IMAGETYPE_PNG, IMAGETYPE_GIF, IMAGETYPE_WEBP])) {
            MensajeFlash::guardarMensaje('El archivo no es una imagen');
            header("Location: index.php?action=modificar_producto&id=$id");
            die();
        }

        // Validar el tamaño del archivo
        if ($_FILES['img']['size'] > 5000000) {
            MensajeFlash::guardarMensaje('El archivo es demasiado grande');
            header("Location: index.php?action=modificar_producto&id=$id");
            die();
        }
    }
}
```

```

$img_name = uniqid() . '_' . $_FILES['img']['name'];
$img_path = 'web/img/' . $img_name;

try {
    // Mover el archivo cargado al servidor
    if (!move_uploaded_file($_FILES['img']['tmp_name'], $img_path)) {
        MensajeFlash::guardarMensaje('Error al guardar la imagen');
        header("Location: index.php?action=modificar_producto&id=$id");
        die();
    }

    //Guardar los datos
    $producto->setNombre($nombre);
    $producto->setDescripcion($descripcion);
    $producto->setCategoria($categoria);
    $producto->setPrecio($precio);
    $producto->setImagen($img_name);

    $productoDAO->actualizar_producto($producto);
    header("Location: index.php?action=ir_modos_admin");
    die();
} catch (Exception $e) {
    // Manejar cualquier excepción que pueda ocurrir durante el proceso de carga de la imagen
    die('Error al guardar la imagen: ' . $e->getMessage());
}
} else {
    $idllaga = $_GET['id'];
    $conn = ConexionBD::conectar();
    $productoDAO = new ProductosDAO($conn);
    $productos = $productoDAO->obtenerProductoPorId($idllaga);
    require 'app/vistas/modificarProductos.php';
}
}

```

La vista de modificar un producto es esta:

Modificar Producto

Seleccionar archivo

Ninguno archivo selec.

Modificar Producto

Si el usuario se equivoca y intenta introducir una imagen que es incorrecta saldrán los mensajes que hemos programado en el método anterior que se encuentra en el controlado, según el error que sea saldrá un mensaje u otro. Por ejemplo, si el usuario introduce un PDF el tipo no es correcto y sale este mensaje:

Modificar Producto

Seleccionar archivo

archivo_.pdf

Modificar Producto

Modificar Producto

El archivo no es una imagen

Seleccionar archivo

Ninguno archivo selec.

Modificar Producto

-Método para insertar productos a la base (modo administrador):

El método es así

```
function insertar_producto() {
    if ($_SERVER['REQUEST_METHOD'] == 'POST') {
        //Recoger datos y filtrarlos
        $nombre = htmlentities($_POST['nombre']);
        $descripcion = htmlentities($_POST['descripcion']);
        $categoria = htmlentities($_POST['categoria']);
        $precio = htmlentities($_POST['precio']);

        // Validar que se haya subido un archivo
        if (!isset($_FILES['img']) || $_FILES['img']['error'] != UPLOAD_ERR_OK) {
            MensajeFlash::guardarMensaje('Error al cargar la imagen');
            header("Location: index.php?action=insertar_producto");
            die();
        }

        //Validar el tipo de archivo
        $img_type = exif_imagetype($_FILES['img']['tmp_name']);
        if (!$img_type || !in_array($img_type, [IMAGETYPE_JPEG, IMAGETYPE_PNG, IMAGETYPE_GIF, IMAGETYPE_WEBP])) {
            MensajeFlash::guardarMensaje('El archivo no es una imagen');
            header("Location: index.php?action=insertar_producto");
            die();
        }

        // Validar el tamaño del archivo
        if ($_FILES['img']['size'] > 5000000) {
            MensajeFlash::guardarMensaje('El archivo es demasiado grande');
            header("Location: index.php?action=insertar_producto");
            die();
        }

        $img_name = uniqid() . '_' . $_FILES['img']['name'];
        $img_path = 'web/img/' . $img_name;

        try {
            // Mover el archivo cargado al servidor
            if (!move_uploaded_file($_FILES['img']['tmp_name'], $img_path)) {
                MensajeFlash::guardarMensaje('Error al guardar la imagen');
                header("Location: index.php?action=insertar_producto");
                die();
            }
        }
    }
}
```

```

//Guardar datos
$producto = new Producto();
$producto->setNombre($nombre);
$producto->setDescripcion($descripcion);
$producto->setCategoria($categoria);
$producto->setPrecio($precio);
$producto->setImagen($img_name);

$productoDAO = new ProductosDAO(ConexionBD::conectar());
$productoDAO->insertar($producto);

header("Location: index.php?action=ir_modos_admin");
die();
} catch (Exception $e) {
    // Manejar cualquier excepción que pueda ocurrir durante el proceso de carga de la imagen
    die("Error al guardar la imagen: " . $e->getMessage());
}
} else {
    require 'app/vistas/insertarProductos.php';
}
}

```

La vista del usuario para insertar un producto es esta

Insertar Producto

Son las mismas comprobaciones que a la hora de modificar algún producto. Por ejemplo:

Insertar Producto

Insertar Producto

El archivo no es una imagen

c. Corrección de errores y optimización del rendimiento

A la hora de realizar la página **GameLoot**, no he tenido muchos errores graves pero cuento algunos errores que me han ocurrido. Al principio tenía algunos errores tontos, como por ejemplo, fallo al inicializar las variables, fallo a la hora de mostrar los productos, a la hora de insertar y modificar, pero muy fáciles de solucionar.

Pero tuve unos problemas a la hora de realizar las funciones de insertar los productos a la lista de favoritos o al carrito. Realicé las funciones usando solo AJAX que se puede hacer utilizando el fetch, pero no me funcionaba y tampoco pude solucionarlo con el postman porque lo intente me daba error también, el resultado es que tuve que cambiar el chip y decir tiene que a ver mas formas de realizarlo. Estuve investigando y busque una solución que era utilizando AJAX y JQuery por lo que decidí probar esa solución y funcionó, y una vez lo hice con favoritos solo era copiarlo para el caso del carrito porque la lógica del script era igual lo único que cambia son las llamadas al controlador ya que cada función, llama a una función diferente en el controlador y utilizan métodos diferentes para realizar dichas funciones correctamente, solo era hacer un par de cambios.

La optimización del rendimiento pienso que es un apartado muy importante que **GameLoot** cumple, ya que esta todo bien optimizado para que funcione todo rápido y el usuario no tenga que esperar a que **GameLoot** cargue. Para optimizar el código y las funcionalidad de **GameLoot** he utilizado AJAX para que la página no recargue cuando el usuario interactué con ella, comprobaciones internas para el tema de seguridad para los datos del usuario, por ejemplo, a principio cuando se insertaban y modificaban los productos, todos los datos que son enviados por el formulario y son recogidos en el controlador se filtran y depuran en el controlador para que no haya problemas, cuando le das a seleccionar una imagen se comprueba los formatos del archivo que se introducen, el tamaño del archivo, etc.

También el código esta optimizado a la hora de hacer comprobaciones, insertar, modificar, etc, utiliza métodos específicos y sencillos para realizar todas esas funciones, como por ejemplo, cuando un usuario inserta un producto a favoritos o al carrito hay métodos que compruebas si ya existe ese producto en favoritos y si es así se pintan los iconos de los botones para dar un estilo mas intuitivo para los usuarios.

V. Documentación a. Documentación técnica


b. Documentación de usuario

Esta documentación de usuario guiará a través de las diferentes funcionalidades y características disponibles en **GameLoot** para que puedan todos los usuarios disfrutar de una experiencia de compra fácil y satisfactoria.

Registro e Inicio de Sesión

Crear una cuenta de usuario

Para registrarte en **GameLoot** y crear una cuenta de usuario, sigue los siguientes pasos:

1. En la página de inicio dale a este icono  que te llevara a la vista de registrarse y iniciar sesión.
2. Completa el formulario de registro proporcionando tu nombre, dirección de correo electrónico y contraseña.
3. Haz clic en el botón de “**Enviar**” para crear la cuenta, te va a iniciar sesión automáticamente.


Iniciar sesión en la plataforma

Si ya tienes cuenta de usuario, puedes iniciar sesión siguiendo estos pasos:

1. En la página de inicio dale a este icono  que te llevara a la vista de registrarse y iniciar sesión.
2. Completa el formulario con tus credenciales y le das al botón “**Enviar**”.

Cerrar Sesión

Los usuarios podrán cerrar su sesión siguiendo los pasos:


1. Una vez hayas iniciado sesión y quieras cerrarla solo te tienes que ir a la parte de arriba de la página en el centro al lado de buscar y saldrá algo como esto **email usuario** 
2. Y tan solo le tendrías que dar un click al icono de la persona y ya cerrarías sesión.

Exploración del catálogo

Búsqueda por título

GameLoot permite buscar videojuegos de acuerdo a diferentes criterios. Para realizar una búsqueda, sigue estos pasos:

1. En la página de principal utiliza el campo de búsqueda en la parte superior de la página e ingresa el título del videojuego que desees buscar.

2. Haz intro o click en el icono de la lupa para obtener los resultados.

Ver detalles del videojuego



Para obtener información sobre un videojuego en particular, sigue estos pasos:

1. Haz click en la imagen del videojuego que te interese.
2. En la página de detalles del videojuego, encontrarás información como la descripción, precio e imágenes.

Carrito de Compras


Añadir videojuegos al carrito

Si deseas comprar un videojuego, añádelo al carrito siguiendo estos pasos:

1. Hay muchas formas de añadir un producto al carrito que son desde la lista de deseos, en los detalles del videojuego y en la página principal donde salen todos los videojuegos, tienes que darle al icono  y se añadirá automáticamente al carrito.
2. Y en la parte de arriba a la derecha de la página sale  este icono al cual si le das te llevará al carrito para ver todos tus productos que has añadido.


Modificar cantidades y eliminar elementos

En el carrito de compras, puedes ajustar las cantidades de los videojuegos o eliminar elementos siguiendo estos pasos:

1. En la página del carrito, verás una lista de los videojuegos que has añadido.
2. Para ajustar la cantidad, puedes modificar este campo
3. Si deseas eliminar un videojuego del carrito, haz click 

Proceder al pago



Una vez que hayas revisado tu carrito de compras y estés listo para realizar el pago, sigue estos pasos:

1. Lo único que tienes que hacer es darle a este botón , y se generará la factura del pedido.

Lista de deseos

Añadir videojuegos a la lista de deseos

Si deseas añadir videojuegos a tu lista de deseos debes seguir los siguientes pasos:

1. Hay varias formas de poder añadir un videojuego a tu lista de deseos como son desde la página principal y desde los detalles del videojuego. 
2. En la parte de arriba a la derecha al lado del carrito en la página principal te vas a encontrar este icono , el cual te llevará a tu lista de deseos donde podrás ver todos tus videojuegos.

Vista del Catálogo

Paginación Catálogo

Los usuarios podéis ver un catálogo bastante extenso pero para verle completamente sigue los siguientes pasos:

1. En la página principal si bajas abajo del todo te vas a encontrar con esto que te va a ir indicando según por que página vayas unos productos y un número diferente según donde te encuentres.
2. Con las flechas que ves puedes ir a la primera página del todo o a la última página del catálogo directamente.



Contacto




Información de la página

Los usuarios tienen acceso a poder ver un poco de información sobre mí abajo del todo de la página donde se encuentran las redes y un apartado que para poder ver toda la información sigue los siguientes pasos:

1. Abajo del todo de la página con un fondo gris encontraras diferentes tipos de información pero la que vamos a explicar en este caso es la opción que pone **“Sobre mí”**.
2. Una vez le des al enlace podrán observar los usuarios una pequeña descripción sobre mí y un pequeño enlace para poder ver y descargar mi currículum. [Ver Currículum](#)

Modo Administrador

Esta opción solo estará disponible para los usuarios que tengan rol administrador que son pocos pero explico como funciona siguiendo los pasos:

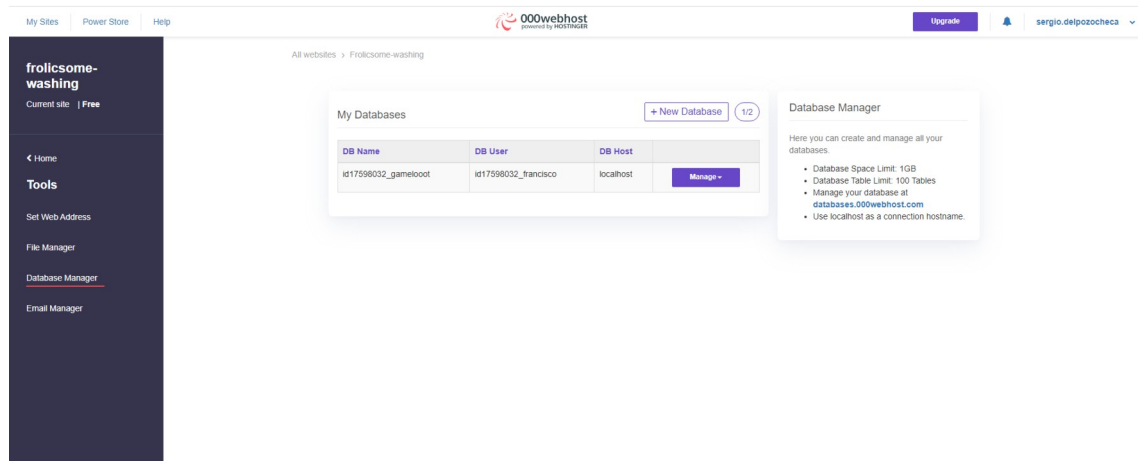
1. Si has iniciado sesión con un usuario con rol administrador en la página principal pegado a la izquierda sale este botón. **Modo administrador**
2. Saldrán todos los videojuegos en una tabla con sus respectivos datos y al final de cada videojuego sale estos iconos para poder borrarlos de la base de datos y modificarlos.  
3. Arriba saldrán dos opciones donde una sirve para insertar productos nuevos a la base de datos y al catálogo. **Insertar Producto**
4. El otro botón justo al lado para ver todos los usuarios que tengan una cuenta **Lista Usuarios**
5. Y dentro del listado de usuarios una opción para ver los respectivos pedidos de cada usuario 

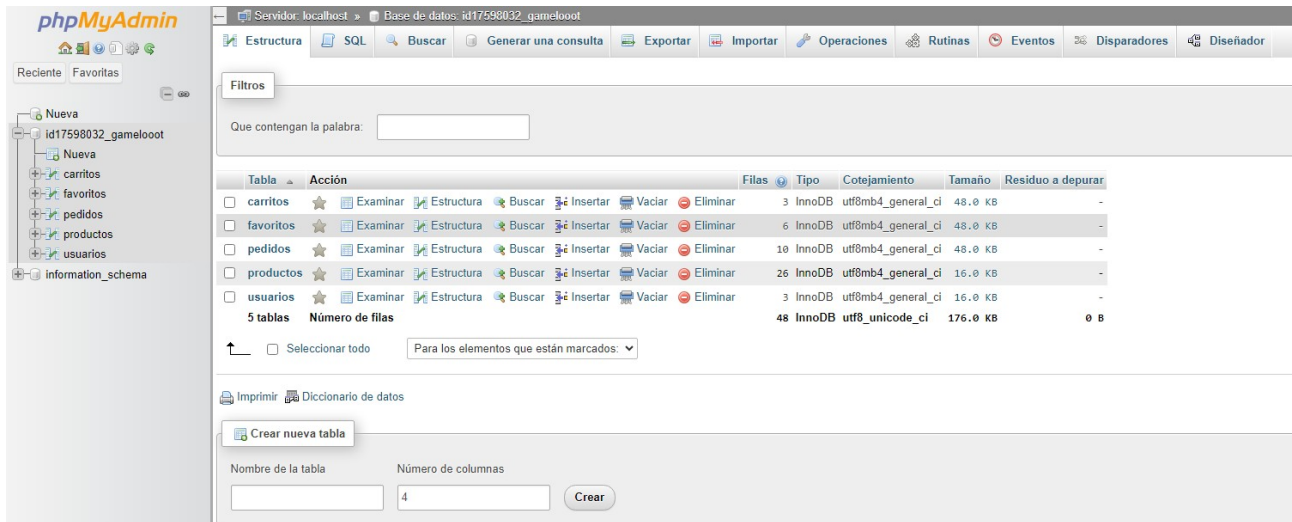
c. Manual de instalación y configuración

Para que todos tengan disponibilidad ya accesibilidad de **GameLoot** utilizaré un hosting, también conocido como un alojamiento web, que en mi caso voy a utilizar 000webhost.

Para poder utilizar 000webhost hay que tener una cuenta creada para poder acceder a sus funciones, os dejo el enlace de 000webhost <https://es.000webhost.com/>.

Primero añadimos la base de datos al host

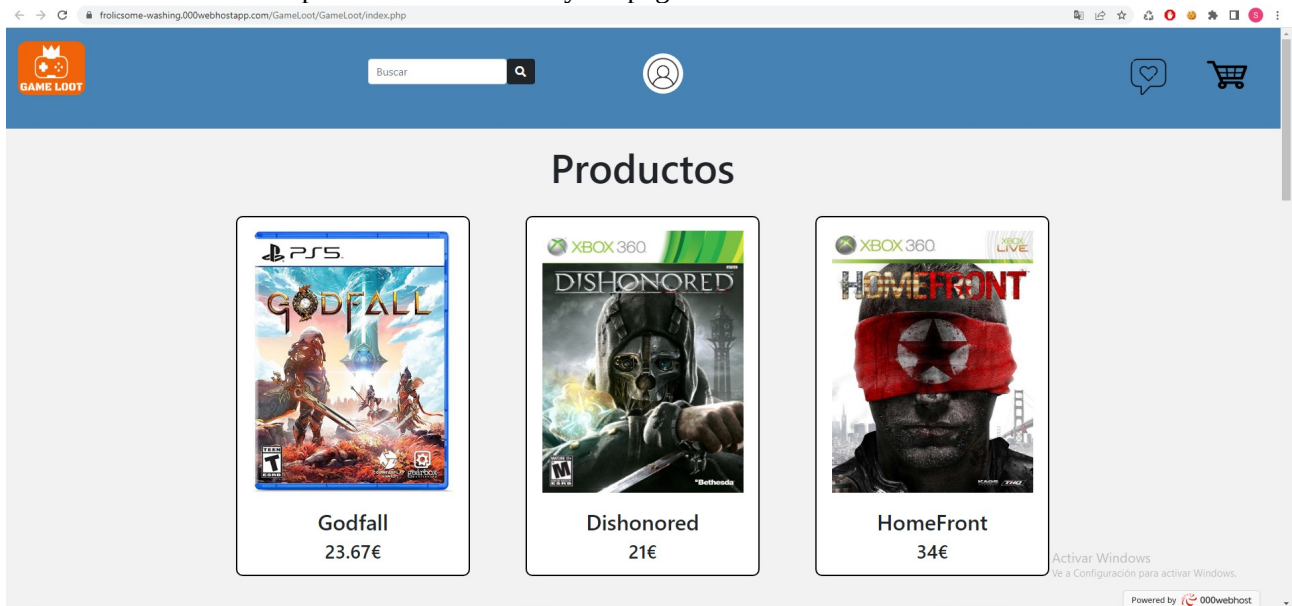




Después creamos una carpeta en public_html para añadir todo el proyecto

<input type="checkbox"/>	GameLoot	2023-06-09 18:26:00	drwxr-xr-x
<input type="checkbox"/>	app	2023-06-09 18:05:00	drwxr-xr-x
<input type="checkbox"/>	librerias	2023-06-09 18:06:00	drwxr-xr-x
<input type="checkbox"/>	nbproject	2023-06-09 18:06:00	drwxr-xr-x
<input type="checkbox"/>	web	2023-06-09 18:34:00	drwxr-xr-x
<input type="checkbox"/>	index.php	4.5 kB	-rw-r--r--

Y cuando le damos a la opción view nos saldrá ya la página **GameLoot**.



VI. Mantenimiento y evolución

a. Plan de mantenimiento y soporte

El plan de mantenimiento y soporte es un apartado esencial para garantizar el correcto funcionamiento y la disponibilidad continua de **GameLoot** y el plan es el siguiente:

1. Actualización de software y seguridad: Regularmente actualizar el software utilizado en **GameLoot**, como el sistema de gestión de contenido (CMS), complementos, temas y componentes. Mantener al día las últimas versiones para aprovechar las mejoras de seguridad y corregir posibles vulnerabilidades.
2. Copias de seguridad regulares: Realizar copias de seguridad periódicas de la base de datos y los archivos de **GameLoot**. Almacenar las copias de seguridad en un lugar seguro y prueba regularmente su restauración para asegurarte de que los datos se puedan recuperar en caso de pérdida o error.
3. Monitoreo y rendimiento: Supervisar el rendimiento de **GameLoot**, incluyendo la velocidad de carga, el tiempo de respuesta del servidor y la disponibilidad.
4. Resolución de problemas: Mantener un proceso de seguimiento y resolución de problemas para abordar cualquier error, fallo o problema técnico que pueda surgir. Establece un sistema de tickets o un canal de comunicación para que los usuarios puedan informar problemas y recibir respuestas rápidas.
5. Soporte a usuarios: Proporcionar un canal de soporte a los usuarios, ya sea a través de chat en vivo o un sistema de tickets. Responder rápidamente a las consultas y dar asistencia técnica y orientación a los usuarios.
6. Seguridad y protección contra ataques: Implementar medidas de seguridad robustas para proteger **GameLoot** contra ataques cibernéticos, como la instalación de certificados SSL, el uso de firewalls, la detección de malwares y la protección contra ataques de fuerza bruta. Realiza auditorías de seguridad periódicas y toma medidas correctivas según sea necesario.
7. Monitorización de tendencias y competencia: Mantenerse al tanto de las tendencias en la industria de los videojuegos y observar a las competencias.

Es importante mantener un enfoque activo y estar atentos a las necesidades de los usuarios para poder dar un servicio de alta calidad y una experiencia positiva en **GameLoot**.

b. Identificación de posibles mejoras y evolución del proyecto

La identificación de posibles mejoras y la evolución del proyecto es un proceso continuo y fundamental para mantener a **GameLoot** actualizada y competitiva. Algunas de las mejoras que se implementarán a **GameLoot** son las siguientes:

1. Encuestas de usuarios: Realizar encuestas y solicitar comentarios directamente a los usuarios de **GameLoot**. Preguntar sobre la experiencia que han tenido a la hora de utilizar la página y comprar los videojuegos, que mejoras les gustaría ver o añadir y cualquier sugerencia que puedan tener.
2. Estilo de la página: Modificar un poco el diseño de la página para que sea mas llamativa y colorida para llamar más la atención de los usuarios.
3. Sistema de servicio técnico: Añadirle a **GameLoot** un sistema de chat en línea con un cuerpo técnico para solucionar y ayudar a los usuarios sobre problemas sobre los videojuegos, pedidos, problemas con la página, etc.
4. Comentarios: Crear un sistema de comentarios para que los usuarios puedan añadir sus opiniones en los detalles de los videojuegos y para que otros usuarios lo vean y puedan compartir diferentes opiniones sobre el videojuego que quieran y que los usuarios puedan interactuar entre sí.
5. Sistema de búsqueda: Modificar el sistema de búsqueda de **GameLoot** para que sea más completo y potente para que los usuarios puedan refinar mas sus búsquedas para encontrar los videojuegos que quieran encontrar.

La evolución de **GameLoot** debe ser continuo y adaptativo. Mantener la página actualizada sobre las tendencias y avances en la industria de los videojuegos y escuchar los comentarios de los usuarios para asegurar que **GameLoot** mantenga relevante y satisfaga las necesidades de los usuarios.

c. Actualizaciones y mejoras futuras

Las actualizaciones que debería hacerse a **GameLoot** para su mayor rendimiento y funcionamiento son las siguientes:

1. Sistema de servicio técnico: Añadirle a **GameLoot** un sistema de chat en línea para poder solucionar y ayudar a los usuarios con cualquier tipo de problema que tenga sobre los pedidos, productos, sobre el funcionamiento de la página, etc.
2. Comentarios en los detalles: Crear la opción de que los usuarios puedan añadir comentarios en los detalles de los videojuegos para poder compartir sus valoraciones y opiniones para que otros usuarios puedan ver las características de cada videojuego y estar seguros de comprarse dicho videojuego.
3. Sistema de búsqueda: Mejorar el sistema de búsqueda para que sea mas potente y robustos para que los usuarios puedan refinar mas sus búsquedas para encontrar mas rápido los videojuegos que están buscando.
4. Personalización de los usuarios: Poder hacer que los usuarios se puedan modificar su propio perfil para dejarlos mas chulos, por ejemplo, que se pudieran cambiar el nombre del usuario, poner y cambiar una foto de perfil que se la verían los demás usuarios, etc.

Son posibles mejoras y actualizaciones futuras que se pueden hacer para que la página sea mas divertida de utilizar y sea mas funcional e interactiva.

VII. Conclusiones

a. Evaluación del proyecto

La validación y evaluación de la página **GameLoot** se llevó a cabo para verificar la funcionalidad y la calidad de la solución desarrollada. Realicé diversas pruebas que permitieron evaluar su rendimiento, la funcionalidad y la usabilidad de la página.

En primer, realicé pruebas de integración para evaluar la interacción entre las distintas funciones y garantizar que funcionaran de manera conjunta de manera adecuada. Durante estas pruebas, se detectaron algunos errores que fueron solucionados. Por otro lado, realicé pruebas de usabilidad simulando usuarios. Recopilé información para poder hacer ajustes en la interfaz de usuario para poder mejorar la experiencia de uso de la página. Simulando usuarios detecté la facilidad de navegación y la claridad de las funcionalidades ofrecidas por la página.

En general, los resultados de la validación y evaluación fueron positivos y confirmaron la funcionalidad y calidad de la página. Las pruebas realizadas demostraron que el sistema cumple con los requisitos funcionales establecidos y se comporta de manera segura y estable.

Sin embargo, también identifiqué algunas áreas de mejora durante la evaluación.

Como conclusión final, la validación y evaluación del proyecto demostraron la funcionalidad de la página desarrollada. Las pruebas realizadas garantizan la seguridad y usabilidad de la página.

b. Cumplimiento de objetivos y requisitos

El principal objetivo de este proyecto era crear una página funcional y segura para todos los usuarios amantes de los videojuegos. Mi objetivo era hacer una página para comprar videojuegos ya sea tanto digitales como videojuegos físicos.

Lo más importante era hacer que el usuario se sintiera a gusto utilizando la página y sea sencilla de utilizar porque no todas las personas se manejan tan bien con las tecnologías, pero yo creo que **GameLoot** es una página muy fácil de utilizar con muchas funcionalidades para que el usuario pueda interactuar con la página, un sistema de búsqueda para encontrar de forma efectiva y rápida los videojuegos que está buscando el cliente, etc.

Se logró integrar un amplio catálogo de videojuegos en la página, permitiendo a los usuarios buscar y explorar diferentes títulos. Además de las funcionalidades básicas, se logró implementar características adicionales que mejoran la experiencia del usuario, como la posibilidad de agregar videojuegos a la lista de favoritos. También el tema de la seguridad y fiabilidad de la página para que todos los usuarios tengan la certeza de que **GameLoot** es bastante seguro ya que se cifran las contraseñas y se depuran y filtran todos los datos enviados por formularios.

En general, se logró desarrollar una página de venta de videojuegos funcional y segura que cumple con los objetivos establecidos. La página proporciona una plataforma segura y fácil de usar para que los usuarios exploren, compren y disfruten de sus videojuegos favoritos.

c. Lecciones aprendidas y recomendaciones para futuros proyectos

Algunas de las lecciones y recomendaciones que he aprendido en el desarrollo de **GameLoot** son:

1. Planificación adecuada: Es fundamental realizar una planificación detallada antes de comenzar el desarrollo de cualquier proyecto. Esto incluye la definición clara de los objetivos, la identificación de los requisitos del sistema y la estimación realista de los tiempos y recursos necesarios. Una planificación adecuada ayuda a evitar retrasos y problemas imprevistos durante el desarrollo.
2. Selección de tecnologías adecuadas: Es importante elegir las tecnologías adecuadas para el desarrollo de la página **GameLoot**. Esto implica investigar y evaluar las diferentes opciones disponibles, teniendo en cuenta factores como la seguridad y la facilidad de uso. La elección de las tecnologías correctas facilita el desarrollo y mejora el rendimiento del sistema.
3. Enfoque en la experiencia del usuario: La experiencia del usuario es fundamental en una página. Es esencial asegurarse de que la interfaz de usuario sea intuitiva, atractiva y fácil de usar.

4. Seguridad de la plataforma: Dado que los datos de los usuarios están involucrados en una página, se debe prestar especial atención a la seguridad de la plataforma. Se deben implementar medidas de seguridad robustas, como el almacenamiento seguro de contraseñas, la protección contra ataques de seguridad y la encriptación de datos sensibles.
5. Mantenimiento y actualizaciones: Una vez que la página esté en funcionamiento, es importante mantenerla y actualizarla regularmente. Esto implica corregir errores, realizar mejoras de rendimiento y seguridad, y agregar nuevas funcionalidades según las necesidades. El mantenimiento continuo garantiza que la página siga siendo segura, eficiente y atractiva para los usuarios.

VIII. Bibliografía y referencias

a. Fuentes utilizadas en el proyecto

Estas son algunas de las fuentes que utilizado para buscar información para poder desarrollar **GameLoot**:

Documentación de tecnologías

1. Documentación de PHP: <https://www.php.net/manual/es/index.php>
2. Documentación de JavaScript: <https://developer.mozilla.org/es/docs/Web/JavaScript>
3. Documentación de HTML: <https://www.w3schools.com/>
4. Documentación de CSS: <https://developer.mozilla.org/es/docs/Web/CSS>
5. Documentación de MySQL: https://docs.oracle.com/cd/E17952_01/index.html
6. Documentación de JQuery: <https://api.jquery.com/>
7. Documentación sobre la librería DOMpdf: https://www.srcodigofuente.es/aprender-php/guia-dompdf-completa?utm_content=cmp-true

b. Referencias y enlaces de interés

Para poder desarrollar **GameLoot** he utilizado algunas páginas para los estilos y algunas dudas sobre PHP algunos métodos que he utilizado y no conocía y para solucionar algunos errores que me han ido surgiendo he utilizado los foros.

Enlaces utilizados

Documentación de PHP: <https://www.php.net/manual/es/index.php>

Documentación sobre bootstrap: <https://getbootstrap.com/>

Para los estilos: <https://www.w3schools.com/>

Foros para solucionar algunos errores: <https://stackoverflow.com/>

Para el tema de las ideas del diseño y como mas o menos organizar la página y que funcionalidad ponerle utilicé como referencia algunas páginas para ver ejemplos como por ejemplo:

MediaMarkt: <https://www.mediamarkt.es/es/checkout>

Eneba: <https://www.eneba.com/es/favorites/wishlist>

Game: <https://www.game.es/>