

GAMERS DEN



Miembros:

- David Cruza Sesmero
- Arturo Fernández López
- Camilo Andrés D'isidoro
- Antonio Bonillo Saiz
- Luis Manuel Pérez Cuadrado
- Jesús Abajo Magro

Contenido

Descripción del proyecto	2
Objetivos que alcanzar	2
Práctica 2	2
Práctica 3	2
Práctica Final	2
Descripción detallada de la aplicación.....	4
Base de datos: Estructura y funcionamiento.....	14
Estructura del código	16
Archivos implementados, explicación de sus funcionalidades.....	17
/css.....	17
/Full calendar.....	17
/Includes/mysql	17
/includes/vistas/común	17
/includes/vistas/plantillas.....	17
/includes.....	17
/.....	20
Usuarios y contraseñas utilizadas.....	23
Uso de JavaScript	23
Uso de VPS.....	24
Tabla de gestión del proyecto	24

Descripción del proyecto

Gamers Den es una página web que se centra sobre todo en un comparador de videojuegos; en dicho comparador estableceremos distintas funcionalidades donde podamos implementar a varios actores (usuario registrado, no registrado, administrador, administrador de foro, etc...), todo esto girará en torno a una base de datos donde se gestione las preferencias de cada usuario y según su rol asignado cada uno de los usuarios tendrá funcionalidades distintas; a su vez implementaremos funciones donde distarán de la idea principal tales como venta de segunda mano, foro, calendario de salida de noticias con noticiero interno, chat “en línea”, etc...

Dentro del proyecto estas funcionalidades están implementadas con varios lenguajes: HTML, PHP y JavaScript.

Objetivos que alcanzar

En esta práctica los objetivos a alcanzar son los siguientes:

Práctica 2

- Detallar la base de datos y su estructura
- Implementar tres funcionalidades: Perfil y chat “en línea”
- Gestión de inicio de sesión y registro de los usuarios
- Implementar distintos roles para los usuarios
- Creación de una memoria para detallar la práctica y nuestro trabajo

Práctica 3

- Reformulación de la base de datos y arreglar errores con hash contraseña
- Implementar funcionalidades de tienda y de noticias
- Acabar con la función de chat e implementación de esta en tienda
- Desarrollar las CSS implementadas en la P2
- Creación de una memoria para detallar la práctica y nuestro trabajo

Práctica Final

- Arreglar distintos bugs o desperfectos de la práctica 3, tales como:
 - **Arreglos comunes:**
 - Indicar el VPS que tenemos asignado

<https://github.com/CDisidoro/Practica2-AW>

- Incluir los usuarios de prueba incluidos (Falta art25)
 - No se liberan recursos `$rs->free ()` cuando se lanza una consulta `SELECT`.
 - Las operaciones de base de datos no escapan (`$conn->real_escape_string ()`) los parámetros del usuario.
 - No se utiliza `HTTP POST` cuando la operación modifica el estado del servidor.
 - Los datos que provienen del usuario no se validan adecuadamente.
 - Las clases de entidad (e.g. Usuario, Mensaje, etc.) generan `HTML`. Las clases de entidad no deben de tener esa responsabilidad.
 - Las operaciones de BD devuelven arrays cuyo contenido son directamente las filas que se obtienen de la base de datos y no instancias de la clase correspondiente.
- **Perfil:**
 - El usuario debe poder subir su propio avatar.
 - En vez de añadir un usuario sin más que se haga una búsqueda del usuario
 - Se envían solicitudes de amistad en vez de añadir amigos sin más.
 - Hacer que al darle al nombre del usuario dirija al perfil además de que se le mande desde el botón del menú.
 - Navegar a los juegos desde la lista de deseos.
 - **Chat:**
 - Refresco de mensajes con JS
 - Se usa un textarea
 - **Tienda:**
 - No se puede acceder a art25.
 - Mejorar el proceso de compra (Al comprar se marcan como reservado, cuando el vendedor envía su juego lo marca como enviado y cuando el comprador lo recibe se marca como recibido).
 - Añadir los juegos que se ha comprado en el perfil.
 - Hacer accesible el carrito en todas las páginas.
 - Al comprar el juego se debe borrar de la tienda.
 - Poner en la lista de juegos un botón de vender copia para simplificar el proceso.
 - Usar `Select2` para simplificar el alta (registro) de un juego a vender.
 - **Noticias:**
 - Dar más espacio para mostrar las noticias.
 - Evitar un overflow de contenido en la página (Quitar el scrollbar de las noticias).
 - No usar números para elegir etiquetas, usar un `select2` con su nombre en su lugar.

- **Juegos:**

- Se muestran repetidas las categorías en los juegos al modificar una categoría.
- Se puede meter un precio negativo a un juego.
- Procesar la descripción para que cuando haya un salto de línea haya un párrafo nuevo en vez de que salga todo apilotonado.

Todos los puntos anteriormente descritos han sido solucionados en dicha entrega. Ahora detallaremos los puntos que hemos abordado en esta entrega final.

- Adaptar estilos de la página web con Bootstrap.
- Calendario funcional donde podemos gestionar eventos.
- Un foro funcional con un sistema de votos de comentarios.
- Retocar la funcionalidad de la tienda referente a la muestra de los productos del usuario.
- Incluir funcionalidad JS en nuestro proyecto en determinadas áreas como el calendario, chat, etc...

Descripción detallada de la aplicación

Cuando el usuario se mete en la aplicación se encontrará con una estructura que se repetirá a lo largo de la página web, la cual se divide en tres contenedores principales que son: Cabecera, footer y contenedor principal.

Cabecera

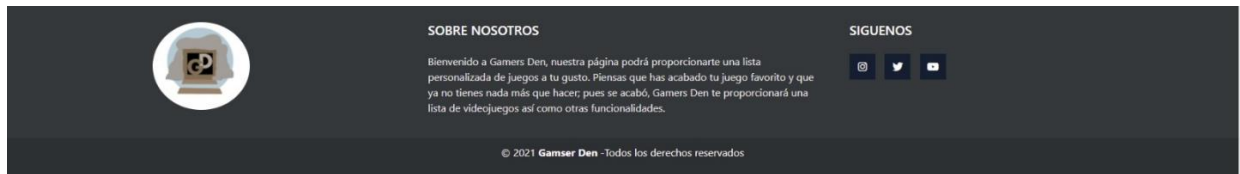


(1) Imagen cabecera Gamers Den

En la cabecera (1) nos encontraremos las principales funcionalidades y ámbitos que podemos encontrar los cuales son: Tienda, Chat, Foro, Noticias, Perfil, Carrito; a parte de ello se puede ver el logo de la página y el nombre del usuario que está logueado en ese momento con su avatar.

Mediante estos botones están enlazadas las distintas funcionalidades que tenemos en nuestra página web, las cuales detallaremos a continuación.

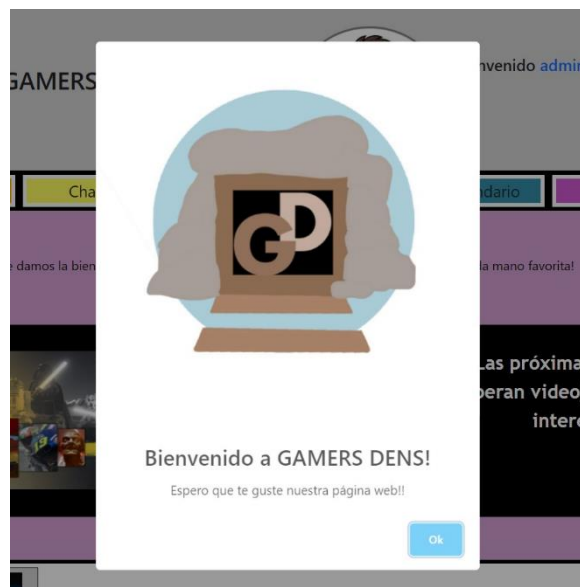
Footer



(2) Imagen del footer

En el footer (Ver figura 2) nos encontraremos una breve descripción de nuestra página web, donde detallamos de una manera rápida para que sirva dicha web, cabe decir que hemos creado las distintas redes sociales a modo de prueba de cada uno de los links que se muestran en la imagen (Instagram, Youtube, Twitter).

Mensaje Emergente



(3) Imagen emergente de Gamers Den

Al meterse un usuario en la página web salta un mensaje emergente () que nos da la bienvenida a la página este mensaje emergente saldrá cada vez que pinchemos sobre el logo de la web.

Página Principal



(4) Imagen de la Página Principal

Nos encontramos una breve introducción a cada uno de los aspectos de nuestra página web (Ver figura 4), donde las imágenes que se van mostrando nos redirigen a la noticia o al aspecto que queramos ver que se nos esté mostrando en ese momento.

Perfil



(5) Imagen de Perfil de usuario

En el perfil del usuario (Ver figura 5) nos encontramos lo siguiente, cabe destacar que no se mostrará el perfil si el usuario no está logueado en nuestra aplicación.

Como podemos ver en la imagen se mostrará el nombre del usuario a su vez podremos ver la imagen (avatar) del usuario que está logueado en ese momento junto con la lista de amigos, donde podremos ver que amigos tiene el usuario y por último acabaremos con la lista de deseos donde podremos ver los juegos que han sido seleccionados por el usuario.

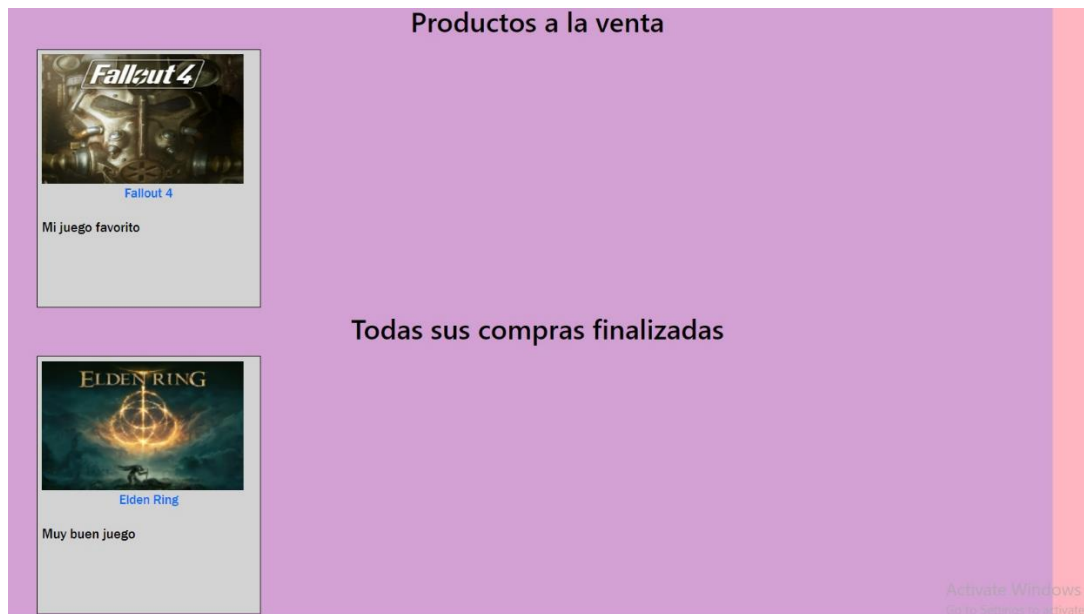
A parte de lo mencionado anteriormente nos encontraremos un enlace a añadir amigos (Ver figura 7) donde podremos rellenar un formulario donde podremos agregar a usuarios como amigos nuestros, también, tenemos cambiar la biografía del usuario (Ver figura 6). Además, podremos ver los productos (Ver figura 8) que le pueden interesar al usuario y los productos que ha comprado hasta el momento.



(6) *Cambia biografía*



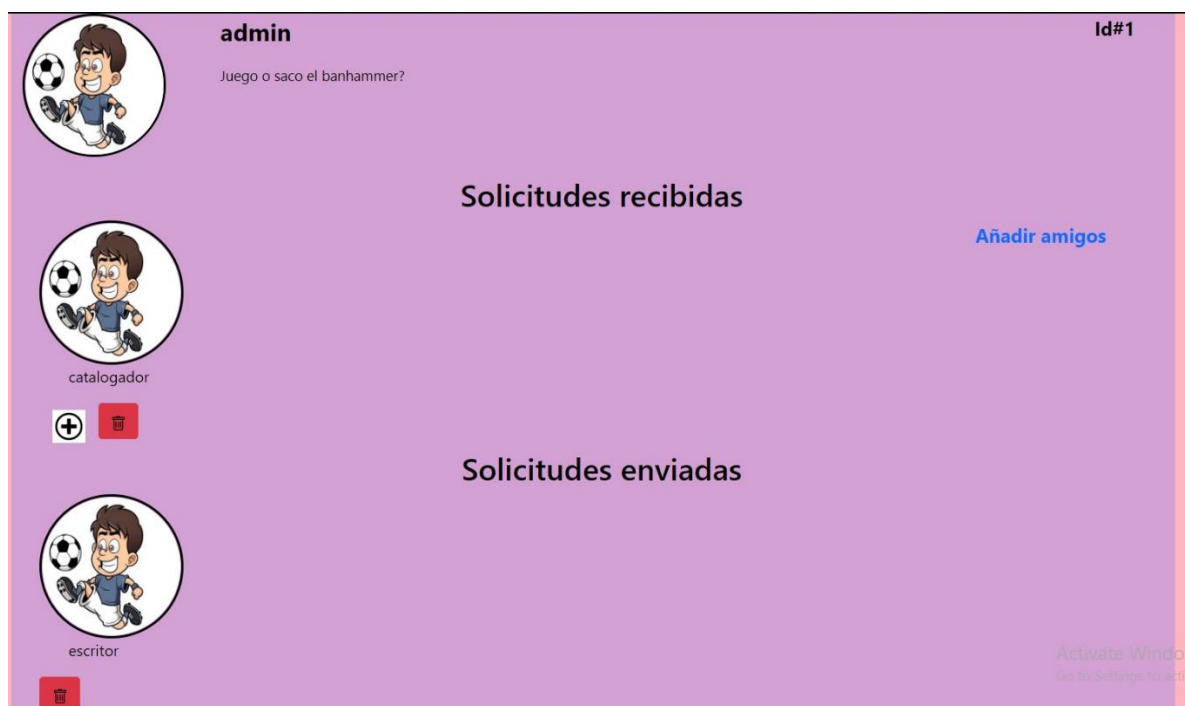
(7) *Imagen de añadir amigo*



(8) Imagen de los productos que están a la venta y de los productos comprados por el usuario

Solicitudes recibidas

Aquí podemos ver las solicitudes que le han llegado o han enviado



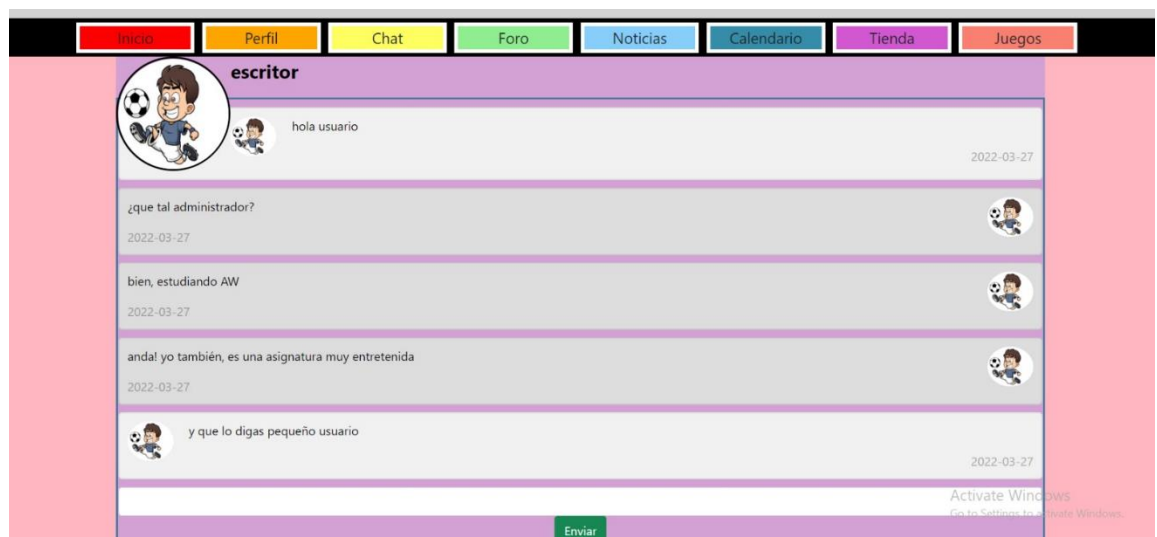
(9) Apartado de solicitudes que ha enviado o recibido un usuario

Chat

Aquí podemos ver los chats que tiene abierto el usuario (Ver figura 10) tanto los que está utilizando para hablar con amigos como para ver los chats que tiene, tras pulsar en uno de ellos veremos lo siguiente (Ver figura 11).



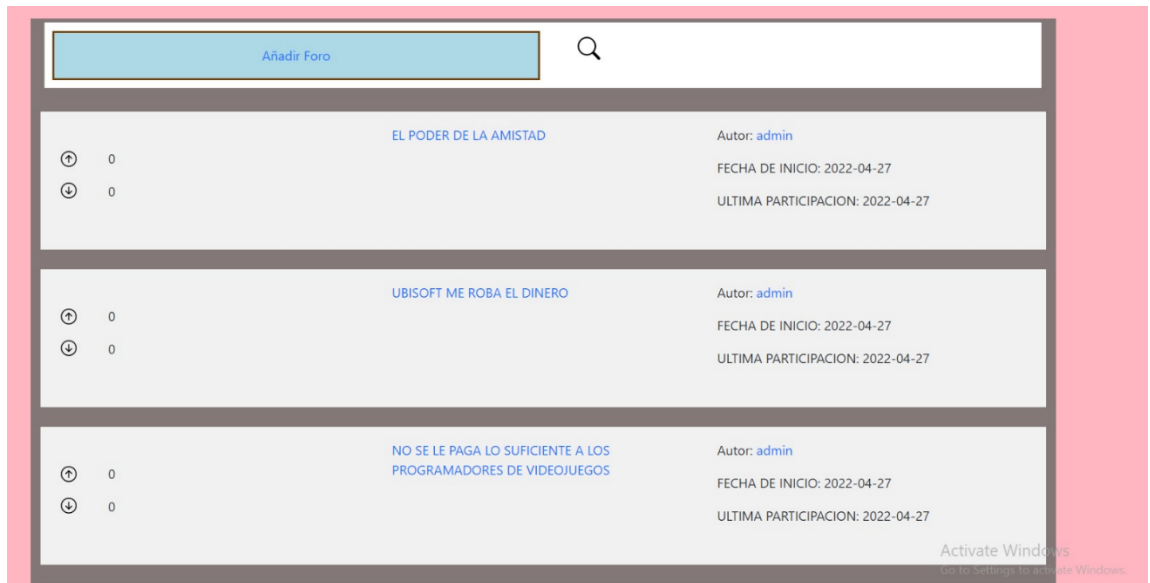
(10) Previsualización de los chats del usuario



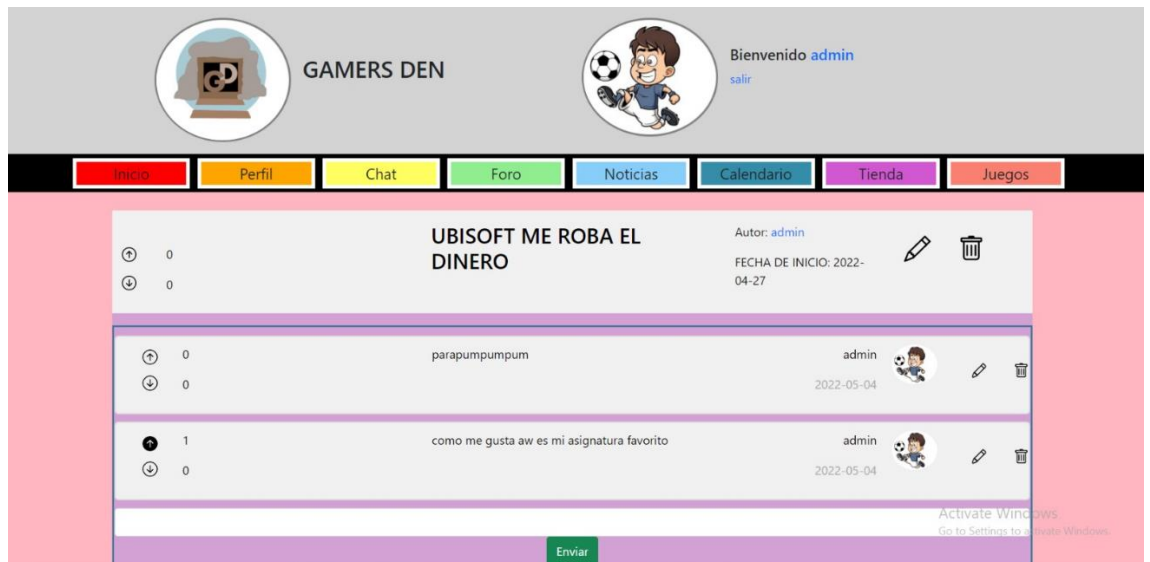
(11) Imagen del chat particular con un usuario

Foro

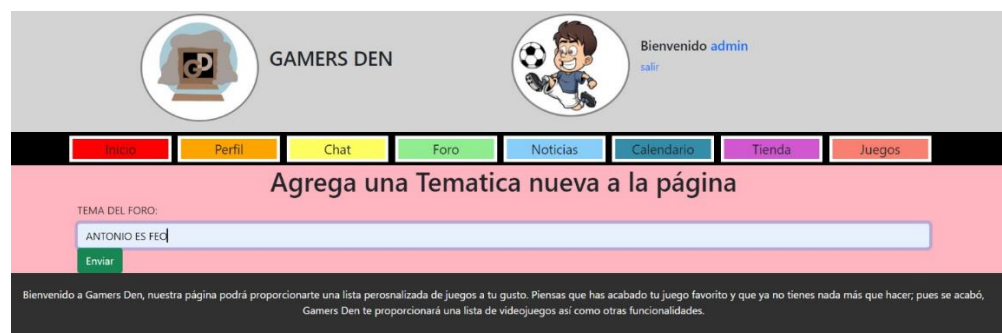
En el foro podremos ver los comentarios que han ido reflejando los usuarios, además de esto tenemos un sistema de votos de comentarios como podemos ver en la imagen (Ver figura 12) y al seleccionar una de ellas nos introduce en el foro particular donde podemos ver el hilo que hemos seleccionado con la información relativa (Ver figura 13), a su vez podemos agregar un nuevo hilo al foro (Ver figura 14).



(12) Foro General



(13) Foro Particular



(14) Añadir un nuevo hilo al foro

Noticia General

Aquí el usuario puede ver todas las noticias (Ver figura 15) que están establecidas dentro de la página web, si pincha en una de ellas pasará a ver la noticia particular (Ver figura 17), donde puede ver todo el contenido de esta y buscar las noticias que queramos (Ver figura 16) y por último cada usuario podrá editar una noticia propia (Ver figura 18).



(15) Noticia General



(16) Busca Noticia



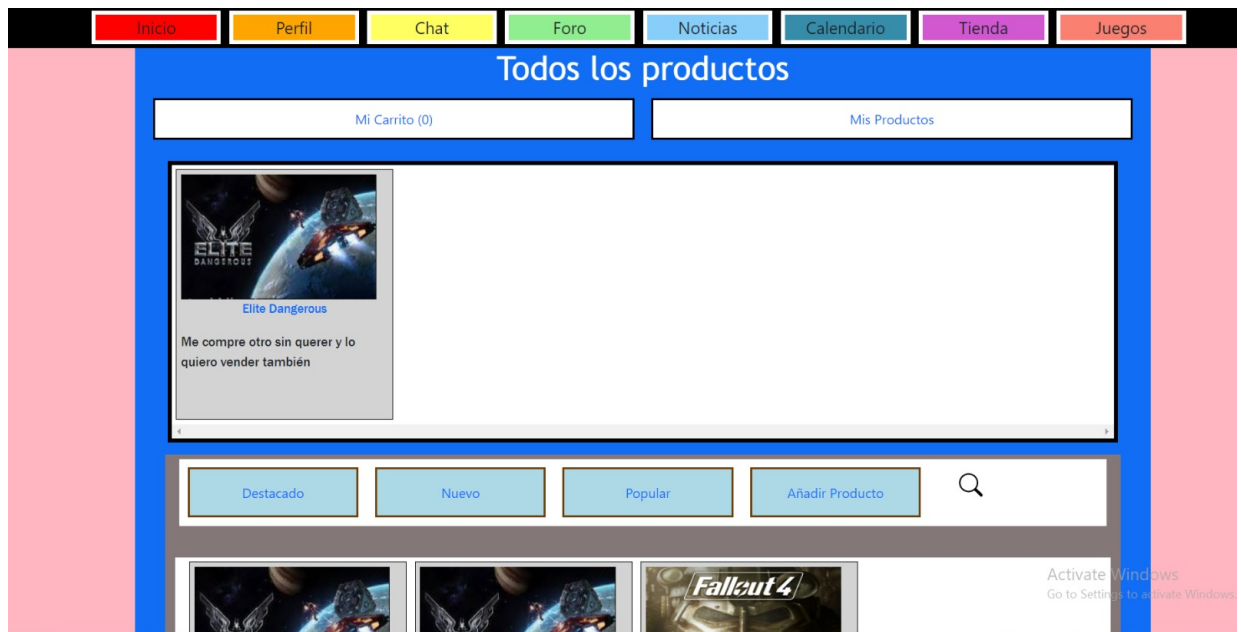
(17) Noticia Particular

 The screenshot shows the edit news form on the Gamers Den website. It includes fields for "Nuevo título:", "Cambiar imagen:" (with a file selection button and "Sin archivos seleccionados" text), "Edita el contenido de la noticia:" (a large text area containing the article text), "Edita la descripción:" (a text field with "Desde DICE han adelantado que están preparando numerosos ajustes en el shooter de Electronic Arts"), and "Cambia la etiqueta:" (a dropdown menu). A green "Enviar" button is at the bottom left.

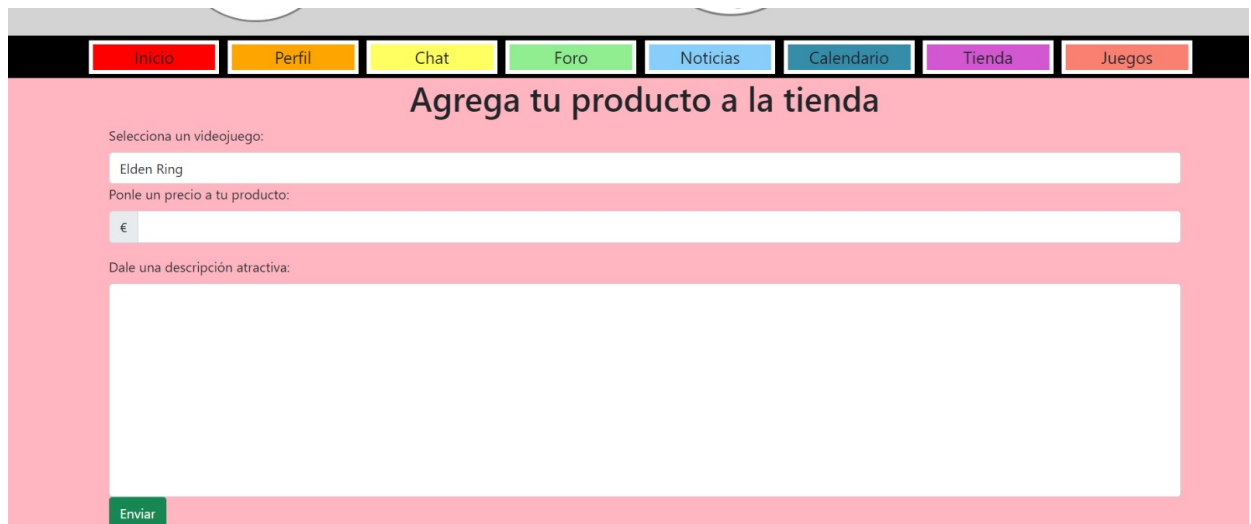
(18) Editar Noticia

Producto

Por último, nos encontramos el apartado de productos donde el usuario puede ver una lista de todos los productos que están ofertados (véase la figura 19), a su vez el propio usuario puede añadir juegos mediante un formulario (véase la figura 20)



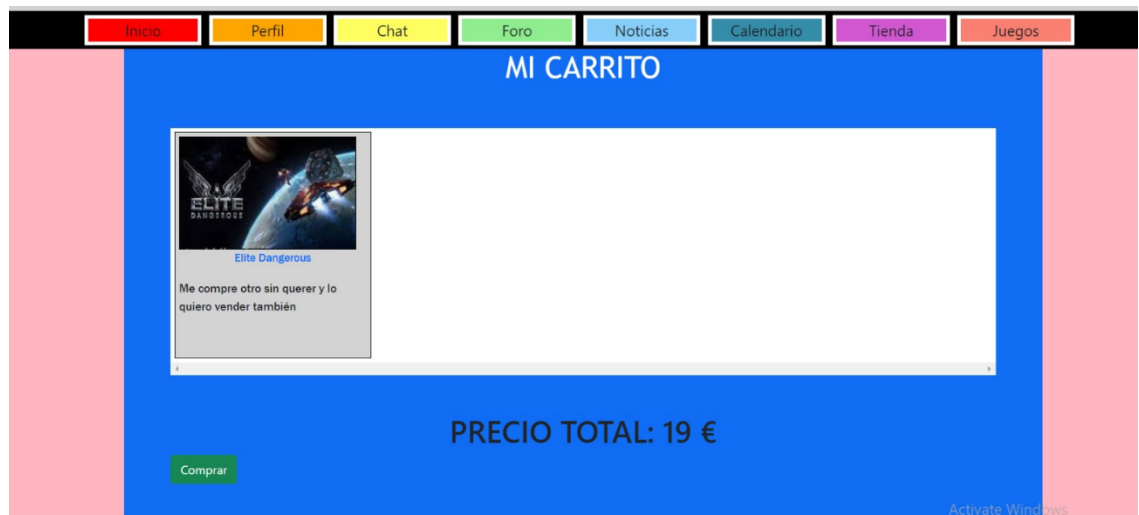
(19) Productos



(20) Agregar producto

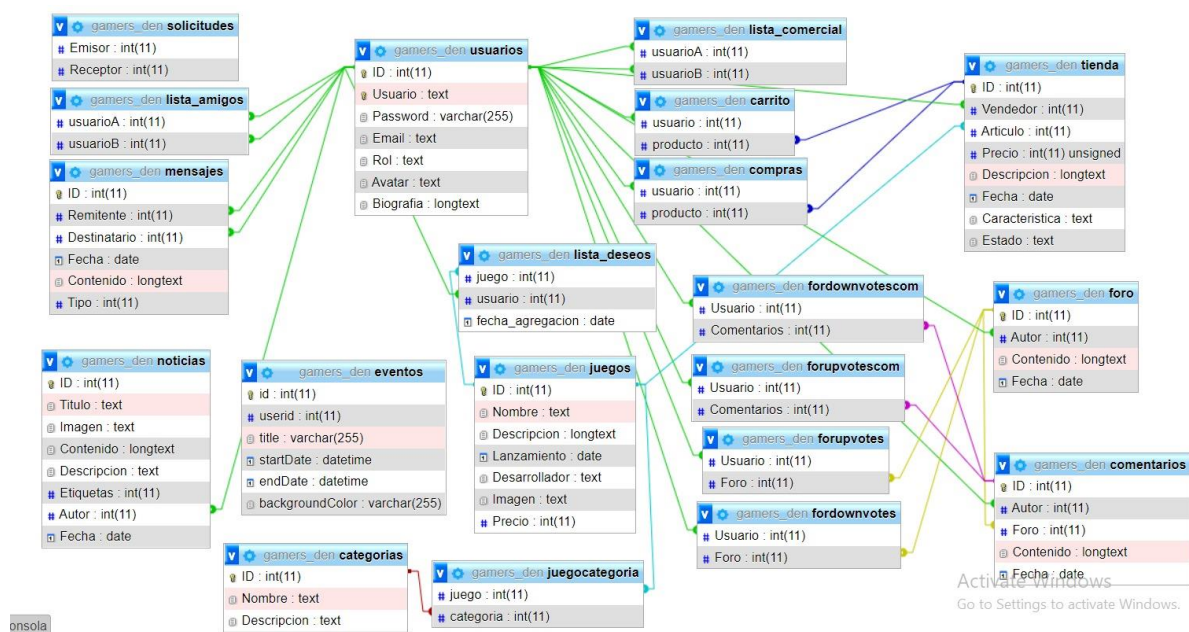
Carrito

En carrito (ver figura 21) podemos ver los productos que ha ido seleccionando el usuario donde a su vez podemos ver el importe de este.



(21) Carrito

Base de datos: Estructura y funcionamiento



En nuestra base de datos para esta etapa del proyecto contamos con 6 tablas, las cuales se detalla su funcionamiento a continuación:

- **Solicitudes:** Aquí tenemos una clase que vincula los extremos de la solicitud tanto del emisor como del receptor.
- **Gestión de votos:** En cada uno de los siguientes que voy a mencionar a continuación nos encontramos la misma estructura, todos relacionan un comentario con un usuario, el cuál está puntuando “negativamente” o “positivamente” un comentario o más bien un hilo que podemos encontrar dentro del foro. Estos son los siguientes: fordwnvotescom, forupvotemcom, forupvotes, fordwnvotes.

- **Comentarios:** Aquí vemos los atributos que utilizamos para la clase comentarios los cuales los utilizamos para el foro y sus atributos por lo general nos sirven para referenciar quién lo ha hecho, de que va, etc...
- **Juegos categoría:** Este lo utilizamos para relacionar un juego a una de las categorías, esto lo hacemos para luego después poder aplicar un filtro durante la búsqueda de los juegos.
- **Eventos:** Aquí podemos ver los atributos que están distribuidos en los eventos que se pueden ir dando dentro de la funcionalidad del calendario.
- **Usuarios:** Esta tabla es el componente principal de la base de datos actualmente. Cada usuario contará con un Identificador en forma de número auto incremental, un nombre de usuario, una contraseña cifrada con un algoritmo SHA, un correo electrónico, un Rol (En el formulario de registro al público cargará por defecto el valor Usuario, y el administrador podrá de dar alta roles privilegiados desde su plataforma de acceso), un número identificador para el avatar (Dispone de 20 posibles opciones para elegir un avatar, con números adicionales para identificar las fotos de perfil de los roles privilegiados), y una Biografía, donde cada usuario podrá personalizar su perfil con una frase o información sobre este.
- **Juegos:** La tabla de juegos es otro componente importante de la BD, aunque actualmente no tendrá aplicación en esta etapa del proyecto. En esta guardaremos la información relativa a los videojuegos que haya en el mercado, como su nombre, desarrolladora, fecha de lanzamiento, descripción, y el precio oficial lanzado por la desarrolladora para este juego. La tabla de juegos está directamente vinculada a la tabla de Tienda, detallada a continuación.
- **Tienda:** Dentro de la tienda contaremos con un ID de la publicación de la tienda, así como un ID que relacione al Vendedor con un ID de la tabla Usuarios, y un ID de producto relacionado a la tabla de Juegos, y el precio que se haya establecido para ese juego.
- **Mensajes:** Esta tabla es bastante importante para la funcionalidad del chat. En esta contamos con un ID único para cada mensaje, y relacionamos con la tabla Usuarios para obtener el ID del remitente y el destinatario. También contamos con un campo para el texto del mensaje, y una fecha de envío que será por defecto un Current Timestamp.
- **Foro:** Esta tabla es fundamental para el funcionamiento del foro y sus comentarios. En esta tendremos un ID único para cada entrada del foro, un ID del autor relacionado a la tabla usuarios, el contenido del foro y la fecha en la que fue creado.
- **Comentarios:** Dentro de esta tabla guardaremos los comentarios que se hayan hecho dentro de una entrada del foro. Para ello tendremos, además de un ID para cada comentario, el ID del autor de ese comentario, y el ID de la entrada del foro al que va relacionado, el contenido de dicho comentario, y la fecha en la que fue realizado.

- **Lista Amigos:** Aquí tendremos una tabla de relación Muchos a Muchos para darle funcionalidad a la lista de amigos (Un usuario A puede tener muchos amigos B, y a su vez, puede ser amigo de muchos usuarios B).
- **Lista Deseos:** En esta tabla tendremos la información relativa a la lista de deseos de los usuarios, siendo de relación muchos a muchos (Un usuario puede tener muchos juegos en su lista de deseos, y un juego puede estar en muchas listas de deseos), incluyendo también la fecha en que fue añadido el juego a esa lista.
- **Noticias:** En este apartado tendremos los atributos necesarios para recoger la información referente a las noticias que se ofrecen dentro de nuestra página web. Los distintos atributos que podemos ver los mostramos en el dibujo.
- **Carrito:** Al usuario cliente se le asignará un producto que haya acordado con un vendedor.
- **Lista comercial:** con esta lista gestionamos los vendedores y clientes que han llegado a un acuerdo por un producto.

Esta Base de datos está sujeta a cambios y modificaciones a lo largo del desarrollo del proyecto, pudiendo implementarse más funcionalidades o eliminar algunas para adaptarlo al producto final

Estructura del código

- **Css:** Aquí encontramos el código relativo al aspecto visual de nuestra página web.
- **FullCalendar:** Archivos relacionados con la funcionalidad del calendario.
- **Img:** Aquí guardamos las imágenes relacionadas con la aplicación.
- **Includes:** carpeta que contiene gran parte del código de la aplicación, en dicha carpeta no puede entrar el usuario debido al htaccess que hemos implementado. Contiene además los ficheros aplicación.php y config.php que se encargan del inicio del programa y acceso a la base de datos y gestión de las rutas de los archivos. Así como los archivos de las clases que implementamos y los distintos formularios que llegamos a manejar dentro de nuestra aplicación.

- Vistas: Esta carpeta contiene a su vez dos subcarpetas las cuales son común (donde tenemos los tres contenedores fundamentales de nuestra página web como pueden ser el sidebar izquierdo, cabecera y el footer) a su vez tenemos la carpeta plantilla donde nos encontramos el archivo plantilla donde gestionamos el contenido de la página web.
- MySQL: contiene los archivos SQL que representan la base de datos de la aplicación.

Archivos implementados, explicación de sus funcionalidades

/css

- **Estilos.css:** Estilos aplicados a la página web

/Full calendar

La mayoría del código que se utiliza viene dado por el material proporcionado por el profesor, cabe destacar que se hacen pequeños cambios en la estructura para adaptarlo a nuestro proyecto.

/Includes/mysql

- Aquí encontramos las tablas referentes a la base de datos, tanto los modelos como tablas rellenas con datos

/includes/vistas/común

- **pie.php:** Breve descripción de la página web.
- **sidebar.php:** Contiene la barra de navegación de la página web, en la que de momento tiene un botón para el Home, para el perfil del usuario y para el chat.
- **cabecera.php:** Muestra el logo de nuestra tienda con un menú para poder navegar por la página, un apartado para iniciar sesión o registrarse en el caso de que no estés registrado y el logo.

/includes/vistas/plantillas

- **plantilla.php:** Es donde vertemos nuestra variable de \$contenidoPrincipal para que se vayan mostrando las funcionalidades

/includes

- **Aplicación.php:** En este archivo gestionamos la clase aplicación donde nos encontramos principalmente las funcionalidades propias para iniciar, cerrar, controlar si está iniciada o establecer una conexión con la base de datos.

- **Categoría.php:** Se establece los atributos propios a la clase de categoría, estableciendo el constructo propio de la clase y sus distintos getters; aparte de ello las distintas funcionalidades propias con `editarCategoría`, `buscarporId`, `borrarCategoría`, etc...
- **Comentario.php:** Aquí nos encontramos la clase comentario donde gestionamos los comentarios que se pueden poner en foro, así como la gestión del voto de estos.
- **Evento.php:** Implementación de la clase json de Evento con sus principales funciones de gestión de esta.
- **Formulario.php:** Clase padre del resto de formularios donde tenemos las funciones de gestión, creación, envío, procesamiento y generación de formularios junto con los atributos principales de los mismos.
- **FormularioAceptarSolicitud.php:** clase hija de formulario donde se gestiona si un usuario puede aceptar una solicitud de amistad enviada por un usuario.
- **FormularioBusquedaForo.php:** Se realiza un formulario que busca un foro mediante una palabra clave.
- **FormularioAmigos.php:** Este archivo hereda de la clase Formulario, en este archivo realizamos un formulario para que el usuario pueda añadir amigos que se encuentren dentro de la base de datos.
- **FormularioCancelarCompra.php:** Formulario donde gestionamos que el usuario pueda cancelar la compra que haya seleccionado en su carrito.
- **FormularioCompraCarrito.php:** Formulario que permite que el usuario pueda gestionar la compra de su carrito.
- **FormularioConfirmaCompra:** Formulario que permite que el usuario pueda gestionar la confirmación de los productos que tiene en su carrito y por tanto la compra.
- **FormularioDownVoteCom.php:** Gestiona cuando una persona vota “negativamente” un comentario, “FormularioDownVoteForo.php” es una clase completamente igual, también hereda de formulario al igual que esta.
- **FormularioUpvoteCom.php:** Gestiona cuando una persona vota “positivamente” un comentario, “FormularioUpvoteForo.php” es una clase completamente igual, también hereda de formulario al igual que esta.
- **FormularioaddListaDeseos.php:** Se establece un formulario en el que principalmente se asocia un usuario con el id de un videojuego para que este sea agregado a una lista de “deseos” donde se refleje que x usuario tiene un interés en x juego.

- **FormularioCambiarAvatar.php:** Establecemos un formulario para que el usuario pueda proporcionar otra imagen para que represente su avatar. Esto mismo nos encontramos con **CambiarBio.php**
- **FormularioBusquedaNoticia.php:** esta clase hereda de **FormularioGestionManual**, en dicho archivo realizamos una búsqueda de una noticia mediante una palabra clave. Esto también se da con **BusquedaJuegos.php** y **BuscarProductos.php**.
- **FormulariosCreación:** Mediante el uso de formularios que extienden de **Formulario** gestionamos distintos formularios asociados a **Negociación, Categoría, Juegos, Foro, Noticia y Producto**.
- **FormulariosEdición:** Mediante el uso de formularios el usuario determinado por su rol podrá editar las clases asociadas a **Negociación, Categoría, Juegos, Foro, Noticia y Producto**.
- **FormularioEliminarAmigos.php:** Formulario para eliminar amigos, utilizamos las funciones correspondientes a usuarios como búsqueda de usuario en la bd y después de comprobar si son amigos dichos usuarios se procede a eliminarlo de la base de datos como amigo. Nos encontramos el mismo caso en los siguientes archivos de eliminación: **Categoría, Deseos, Juego, Foro, Comentarios, solicitud, solicitud entrante, Noticia y Producto**.
- **FormularioGestionManual.php:** esta clase extiende de **Formulario** y lo utilizamos para que nos devuelva después del envío de un formulario una clase de **resultadoGestionFormulario**.
- **FormularioLogin.php:** Formulario que comprueba si los datos introducidos de un usuario coinciden con los que hay en la base de datos y por tanto te permite hacer login en la app.
- **FormularioMandaMensajes.php:** Establece un formulario para enviar mensajes donde antes de nada chequea si el destinatario de dicho mensaje es amigo con el que envía el mensaje a su vez se utilizan clases correspondientes a **Usuario** como **addMensaje** donde ya todo esto va a formar parte del chat. Cabe resaltar que este archivo se divide en dos ámbitos: **Amigo** y **Vendedor**, según si se trata de un chat normal con un amigo o establecemos una negociación con un vendedor.
- **FormularioRegistro.php:** Establecemos un formulario de registro donde le pedimos al usuario que nos proporcione un nombre de usuario, un email y una contraseña la cuál se

le añadirá una doble verificación para que no haya errores a la hora de introducir la contraseña.

- **Mensaje.php:** Establece la clase mensaje proporcionando las distintas funcionalidades que hacen que se retorne los distintos atributos necesarios de la clase.
- **Noticia.php:** Aquí tenemos la clase noticia donde gestionamos los atributos de esta, así como mostramos las distintas funciones que vamos a utilizar para con ella; tales como: `subeNoticia`, `cargarNoticia`, `buscaNoticia`, `cargarCaracterísticas` y los distintos getters que necesitamos para recoger los atributos más directos de la misma.
- **Producto.php:** clase que especifica los atributos propios de producto, esta clase es muy parecida a videojuegos únicamente que esa clase se utiliza para referenciar los videojuegos que tiene un usuario en su perfil y esta se utiliza para ver todos los videojuegos disponibles para los usuarios, así como funciones para subir productos a la tienda o enseñar por características.
- **ResultadoGestionFormulario.php:** Aquí tenemos los resultados propios de la gestión de los formularios como si se ha enviado con éxito, si ha habido errores a la hora de procesarlo o los resultados aparentes que saca dichos formularios.
- **Usuario.php:** En la clase usuario gestionamos los atributos principales del usuario, desde sus datos a los roles que puede tener el usuario.
- **Videojuegos.php:** Clase que sirve para mostrar los atributos de los videojuegos de los cuales dispone el usuario en su perfil, tiene un matiz distinto a productos porque estos son los que tiene referenciado en sus listas.
- **Config.php:** Tiene los parámetros de configuración básicos para el funcionamiento de la página, como el charset, el inicio de una sesión, y la conexión a la base de datos.

./

- **addAmigo.php:** Nos proporciona un formulario que proviene de la clase `formularioAmigos`.

- **CalendarioPrincipal.php:** Gestión en la variable de contenido principal del calendario, hace la misma función que una plantilla.
- **Bienvenido.php:** Establecemos una página de bienvenida mediante un pequeño script que a la vez muestra nuestro logo.
- **buscarNoticia.php:** En este archivo utilizamos el formulario 'FormularioBusquedaNoticia' donde buscamos una noticia mediante una palabra clave.
- **buscarJuego.php:** Inicia la gestión del formulario búsqueda de videojuegos.
- **buscarProducto.php:** Inicia la gestión del formulario búsqueda de productos.
- **chat.php:** En este archivo se gestiona el chat desde su aspecto a como gestionamos con el set Interval () que emula que el chat es en línea puesto que establecemos un refresco.
- **Chat_aux.php:** En este archivo lo que hacemos es volcar lo que hemos recogido en el archivo de log en la base de datos en el campo mensajes a su vez mostramos el chat particular del usuario que esté utilizando dicha función en ese momento junto con el avatar y nombre del destinatario.
- **Chat_amigo.php:** genera un chat mediante formularioMandaMensajesAmigo reconoce la lista de amigos del usuario selecciona el id del usuario amigo y le proporciona los mensajes correspondientes a su chat.
- **Chat_negocio.php:** mediante las funciones dadas por chat auxiliar, establecemos esta vez un chat con un vendedor para negociar sobre un producto.
- **crearX.php:** En cada uno de estos documentos: Categoría, Productos, Foro, Mensaje, Noticia y Juego creamos instancias de cada uno de estos objetos los cuales se gestionan con los distintos formularios de creación que dispone.
- **editarX.php:** Formularios de edición de las categorías anteriormente mencionadas.
- **Juego_particular.php:** presenta las funciones necesarias para la gestión de los juegos particulares del usuario, así como la gestión y funcionalidad de los distintos botones como edición o creación.
- **eliminarAmigo.php:** Utilización de 'FormularioEliminarAmigos'; básicamente hacemos un delete de las tablas correspondientes básicamente de lista de amigos
- **cambioAvatar.php:** formulario para cambiar el avatar del usuario.

- **cambioBio.php:** formulario para cambiar la biografía del usuario.
- **inbox.php:** generación de funciones y contenido para chat particular tales como generarAvatar o generarChat.
- **index.php:** página principal de Gamers Deen.
- **mostrarBusquedaNoticias.php:** capturamos en la variable contenido principal el formulario de búsqueda de noticia, así como mostramos las noticias que hay disponibles en ese momento en la base de datos.
- **mostrarBuscarJuego.php:** capturamos en la variable contenido principal el formulario de búsqueda de juego, así como mostramos los juegos que hay disponibles en ese momento en la base de datos.
- **Notfound.php:** Es un archivo donde se muestra una imagen de un gato para denotar que esa página está en construcción o todavía no se ha diseñado.
- **Noticias_concretas.php:** Búsqueda de noticia individual, esto se hace mediante una función llamada buscaNoticia que se encuentra dentro de Noticia.php.
- **Noticias_principal.php:** aquí mostramos las noticias principales divididas por tags o las características que disponen las mismas.
- **Noticias_aux.php:** Se muestra mediante la función de generaContenidoPrincipal el contenedor de noticias junto con las distintas secciones de noticias (Nueva, destacada, popular), función de búsqueda y un cuadro de noticias donde las establecemos.
- **Tienda.php:** Nuestra tienda donde se exponen los productos ('Videojuegos') mediante una disposición de rejilla y a su vez las imágenes de los juegos y sus propios nombres tienen un enlace que deben redirigirte a tienda_particular.php.

- **Tienda_particular.php:** es parecido al chat particular, una vez has seleccionado un producto te lleva a una página donde se muestran los datos de dicho producto, así como proporcionar un botón que te permita establecer un chat con el vendedor.
- **registro.php:** Muestra la interfaz para poder registrarse.
- **login.php:** Presenta la funcionalidad necesaria para logear a un usuario con control de errores.
- **logout.php:** Funcionalidad necesaria para hacer el logout de un usuario de nuestra app.
- **Perfil.php:** Aquí se mostrará toda la información del usuario que haya iniciado sesión, así como su lista de amigos, su foto de perfil, y lista de deseos.
- **Perfil_aux.php:** Se establecen distintas funcionalidades útiles para la generación del perfil del usuario como su lista de amigos, su foto de perfil, y lista de deseos, la gestión del contenido principal donde se muestra dichos resultados.
- **Perfil_ext.php:** Sirve para ver el perfil de un usuario sin acceder a todo lo relativo a perfil.
- **CargaComentario.php:** Gestiona la carga de los comentarios junto con los botones referentes al mismo, ocurre lo mismo con **Foro y mensajes**.
- **Carrito.php:** Aquí gestionamos los arrays referentes a los productos que ha ido seleccionando el usuario.

Usuarios y contraseñas utilizadas

Usuarios	Contraseñas
Admin	adminpass
Escritor	escritorpass
Catalogador	catalogapass
Moderador	modpass
Usuario	userpass
Art25	12345

Uso de JavaScript

Hemos utilizado JavaScript para facilitar el uso de distintas funcionalidades como puede ser la gestión de envío de mensajes entre usuarios dada en el **chat** donde mediante peticiones el script lo que hace es enviar una solicitud AJAX mediante el uso del objeto XMLHttpRequest, haciendo las comprobaciones pertinentes; hacemos los controles de los mensajes mediante las siguientes variables: onreadystatechange nos

sirve para controlar la respuesta del servidor, y (`this.readyState == 4 && this.status == 200`) significa que la respuesta del servidor está lista para ser procesada, por eso sólo cuando esta condición se cumple es cuando procedemos a enviar el mensaje del usuario y conseguimos refrescar únicamente el área de texto donde mostramos los mensajes del usuario lo que nos permite mantener en todo momento lo que está escribiendo el usuario puesto que lo que escribía anteriormente se borraba debido a la función `set Interval` la cual producía que se refrescase la página entera con lo que eso implicaba lo que estaba escribiendo nuestro usuario.

También hemos utilizado JavaScript en **full calendar** para añadir, editar y eliminar eventos lo que hemos hecho ha sido que montan en el añadido evento y hemos creado un modal y a partir de ahí el usuario puede introducir los datos luego con Ajax estos envían al servidor y se procesan y si todo está bien se añade el evento al calendario.

El **mensaje emergente del index** también se realiza con JavaScript, por último, el sistema de votos también se refresca como chat mediante la gestión de peticiones con Ajax dando una sensación de refresco constante.

Uso de VPS

Nuestro VPS asignado es el 05 (se accede a través de `vm05.containers.fdi.ucm.es`).

Para poder gestionar la aplicación web debemos acceder a Apache Guacamole, e introducir las credenciales de acceso pertinentes. Desde ahí se puede subir los ficheros del proyecto a la carpeta `/var/www`, en la cual tendremos una página para producción y otra para pruebas (`vm05-beta.containers...`).

Y para la gestión de la BD accederemos a `phpmyadmin.containers.fdi.ucm.es` y usaremos las mismas credenciales para acceder a Guacamole, seleccionando el servidor asignado al VPS (`vm05.db.swarm.test`). Desde aquí la gestión de la BD es idéntica a como se gestiona en local a través de XAMPP

Tabla de gestión del proyecto

Aquí mostraremos de una manera rápida la asignación de los miembros del equipo a las distintas funcionalidades propuestas; para ver la “carga de trabajo” de cada alumno esta se mostrará en GitHub mediante la pestaña *Insights/Contributors*, cabe resaltar que muchas veces como hacemos por parejas una determinada tarea una persona es la encargada de subir dicho código, así que el “esfuerzo” será detallado por cada miembro del grupo con las encuestas que se proporcionan el día de la entrega de cada práctica, normalmente esta carga de trabajo suele estar distribuida y se están cumpliendo los objetivos de cada uno.

Todo el contenido del proyecto ha sido revisado por todo el conjunto de miembros.

El enlace a nuestra carpeta de GitHub se muestra en el pie de página.

Práctica Final

Mejoras en el chat	<ul style="list-style-type: none"> • David Cruza Sesmero • Arturo López Fernández • Camilo Andrés D'isidoro
Foro	<ul style="list-style-type: none"> • David Cruza Sesmero • Arturo López Fernández • Camilo Andrés D'isidoro
Foro Particular	<ul style="list-style-type: none"> • Arturo López Fernández
Memoria	<ul style="list-style-type: none"> • David Cruza Sesmero
Calendario	<ul style="list-style-type: none"> • Antonio Bonillo Saiz • Jesús Abajo Magro • Luis Manuel Pérez Cuadrado
Arreglos sobre la práctica 3 <i>(Todos hemos realizado contribuciones en la corrección del feedback de la p3)</i>	<ul style="list-style-type: none"> • David Cruza Sesmero • Arturo López Fernández • Camilo Andrés D'isidoro • Antonio Bonillo Saiz • Jesús Abajo Magro • Luis Manuel Pérez Cuadrado
Bootstrap CSS	<ul style="list-style-type: none"> • Camilo Andrés D'isidoro
Mantenimiento del contenedor VPS	<ul style="list-style-type: none"> • Camilo Andrés D'isidoro

Práctica 3

Desplegar aplicación en VPS	<ul style="list-style-type: none"> • Camilo Andrés D'isidoro
Tienda funcional	<ul style="list-style-type: none"> • David Cruza Sesmero • Arturo López Fernández • Camilo Andrés D'isidoro
Tienda particular	<ul style="list-style-type: none"> • David Cruza Sesmero • Arturo López Fernández • Camilo Andrés D'isidoro
Chat e implementación de este	<ul style="list-style-type: none"> • David Cruza Sesmero • Arturo López Fernández

Noticias	<ul style="list-style-type: none"> • Antonio Bonillo Saiz • Jesús Abajo Magro • Luis Manuel Pérez Cuadrado
Noticias particulares	<ul style="list-style-type: none"> • Antonio Bonillo Saiz • Jesús Abajo Magro • Luis Manuel Pérez Cuadrado
Estilos CSS mejoras	<ul style="list-style-type: none"> • Arturo López Fernández • Antonio Bonillo Saiz • David Cruza Sesmero (Solo rejilla)
Memoria práctica 3	<ul style="list-style-type: none"> • Camilo Andrés D'isidoro • David Cruza Sesmero
Implementar/Arreglar funciones perfil usuario	<ul style="list-style-type: none"> • Arturo López Fernández • Antonio Bonillo Saiz

Práctica 2

Layout funcional	<ul style="list-style-type: none"> • Luis Manuel Pérez Cuadrado • Arturo López Fernández
Base de datos	<ul style="list-style-type: none"> • Camilo Andrés D'isidoro • David Cruza Sesmero
Procesamiento login usuario	<ul style="list-style-type: none"> • David Cruza Sesmero • Camilo Andrés D'isidoro
Registro	<ul style="list-style-type: none"> • David Cruza Sesmero • Camilo Andrés D'isidoro
Chat “en línea”	<ul style="list-style-type: none"> • David Cruza Sesmero • Arturo López Fernández
Creación Perfil Usuario	<ul style="list-style-type: none"> • Antonio Bonillo Saiz • Jesús Abajo Magro
Creación y desarrollo de la memoria	<ul style="list-style-type: none"> • David Cruza Sesmero • Camilo Andrés D'isidoro

Especificación individual de las actividades

Para mostrar un mayor detalle sobre la gestión del proyecto mostramos a continuación dentro de las asignaciones de las tareas el grado de aportación en cada uno de los ámbitos abordados.

Arturo López Fernández	<ul style="list-style-type: none"> • Creación del layout principal. • Desarrollador de chat (general y particular), principal desarrollador de chat auxiliar, aplicación de filtros de búsqueda de usuarios tanto para ventas como individual. • Desarrollador principal de la funcionalidad de foro junto con el sistema de votos. • Creación del carrito del Usuario y compra de productos. • Asignación de productos en el perfil del usuario. • Mantenimiento de JS y PHP. • Desarrollo de la funcionalidad de tienda, gestión de listas de deseos y juegos y aplicación de filtros de búsqueda. • Gestión solicitudes de amistad. • Aportaciones en noticias. • Depuración de bugs.
Antonio Bonillo Saiz	<ul style="list-style-type: none"> • Principal desarrollador de la gestión de perfil de usuario. • Principal desarrollador de la funcionalidad de calendario. • Aportes en tienda. • Principal desarrollador de las CSS del proyecto. • Principal desarrollador de noticias. • Implementación de JS en el sistema de gestión de eventos de calendario.
Jesús Abajo Magro	<ul style="list-style-type: none"> • Desarrollador del perfil de usuario y funcionalidades. • Desarrollador y mantenimiento de calendario, ayuda con JS. • Principal creador de las CSS del proyecto. • Desarrollo de noticias. • Mantenimiento de PHP y CSS.

David Cruza Sesmero	<ul style="list-style-type: none">• Creación del login, logout y registro de usuario.• Implementación de las clases de usuario.• Desarrollo de formularios de edición, eliminación y creación (Ej Usuarios, Tienda).• Creación y gestión de la aplicación con la base de datos.• Aportes en la funcionalidad de chat general y particular, a su vez tratamiento del JS en el mismo.• Desarrollo de la Juegos (Clases principales y su gestión) y aportación de código css a la misma.• Correcciones sobre Perfil de Usuario.• Recopilación de todas las memorias de las prácticas.• Mantenimiento de PHP y CSS.
Camilo Andrés D'isidoro	<ul style="list-style-type: none">• Creación del login, logout y registro de usuario.• Desarrollador de funcionalidades en tienda, perfil y chat.• Desarrollo de cambiarAvatar y cambiarBio.• Desarrollo de lista deseos, lista juegos.• Desarrollo de tienda.• Desarrollo de funcionalidad JS en chat.• Gestión del contenedor VPS y control de bugs.• Paso de CSS a Bootstrap.• Mantenimiento de PHP, CSS y JS.• Aportación en la memoria (ámbito base de datos).• Reorganización del código.
Luis Manuel Pérez Cuadrado	<ul style="list-style-type: none">• Creación del layout principal• Aportaciones en la gestión del usuario en perfil y noticias.• Mantenimiento de errores en PHP.• Aportaciones en CSS.

	<ul style="list-style-type: none">• Desarrollo de calendario y aportación de funcionalidades JS en el mismo.
--	--