

TÍTULO DE PROYECTO

Antecedentes

Este proyecto muestra todos los pasos a seguir para recolectar, organizar, tratar y construir un portal Web empezando desde cero, para montar una página web como si fuera un “foro” para comentar diversos temas y valorar sobre ellos. He optado por un diseño más gráfico que de texto ya que por experiencia personal un foro normal llega a ser tedioso si hay demasiadas páginas junto al texto (como la combinación de colores negros con letra blanca que acaba cansando la vista), por eso prefiero organizarlo por imágenes, facilitando la búsqueda del tema y ver el texto más cómodamente.

Se han utilizado distintos lenguajes de programación, como pueden ser PHP y JavaScript para su desarrollo, y para su diseño se ha hecho uso de estilos CSS y de librerías como Boostrap que nos facilita la creación de diseño responsive, columnas o cualquier recurso que necesitemos.

REQUISITOS

Queremos crear una página que mantenga la esencia de un foro que nos permita publicar, comentar y realizar acciones en ella de manera sencilla para la mayoría de los usuarios.

También queremos permitir que el Usuario pueda filtrar sus publicaciones, valoraciones o comentarios que ha realizado para encontrarlos rápidamente y poder realizar modificaciones o eliminaciones.

Requisitos funcionales

Vamos a crear una página clara de noticias que nos permita ver, comentar y realizar acciones en ella de manera sencilla para todos los usuarios.

También queremos permitir que el Usuario pueda ver las noticias más comentadas y pueda interactuar con ellas, así como comentándolas y dando su opinión sobre que les parece.

1. El usuario se podrá registrar y loguear.
2. Si el usuario no cierra la sesión se queda logueada, es decir hasta que no cierra la sesión no sale.
3. Solo los usuarios registrados pueden ver la noticia entera y comentar en ella.
4. Los usuarios no registrados solo podrán ver el título de la noticia y mitad de la descripción.
5. El usuario registrado tendrá la opción de modificar su perfil, también tendrá la opción de recuperar la contraseña, si no se acuerda de ella y eliminar su perfil.
6. El usuario registrado podrá ver las noticias mas comentadas.
7. Los administradores tienen un panel de administración.
8. Los administradores pueden ver los datos de los usuarios, cambiar su rol y poder eliminarlos.
9. Solo el administrador podrá hacer a un usuario normal admin.
10. Los administradores pueden ver las plataformas que hay en la página, insertar nuevas plataformas, modificarlas y eliminarlas.
11. Los administradores pueden ver las noticias que hay en la página, insertar nuevas noticias, modificarlas y eliminarlas.
12. Los administradores pueden ver los comentarios, aceptarlos o eliminarlos.
13. La página tendrá diferentes secciones de noticias (PC, Nintendo, Playstation, Xbox).
14. La página principal tendrá un carrusel en el que se muestran las consolas, estas se

actualizan conforme se vaya actualizando la página.

15. Para acceder a las noticias tendrán que clicar en una de las tarjetas y le mostrará toda la información aportada.

16. Diseño responsive en todas las páginas.

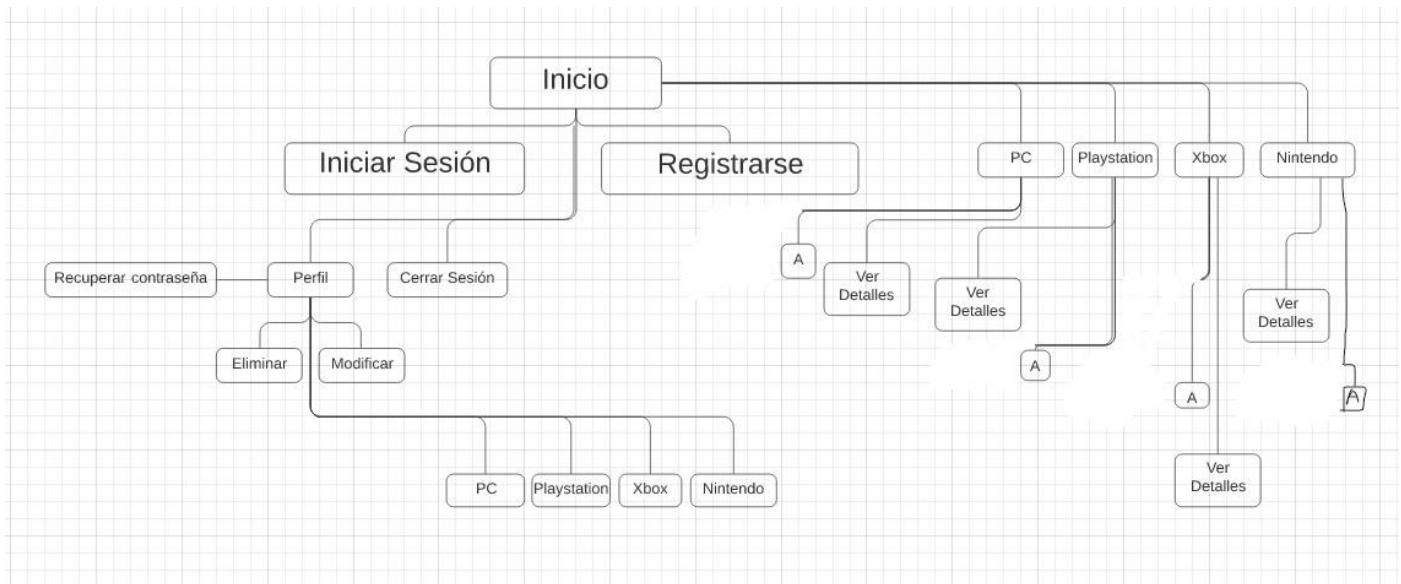
17. Control de errores:

- Usuario
 - Nombre
 - Password
 - Apellido1
 - Apellido2
 - Email
 - Dirección
 - DNI
 - Teléfono
 - Código Postal
 - Provincia (elegible desde un despegable)
 - Población (se llenará según los datos de la provincia seleccionada)

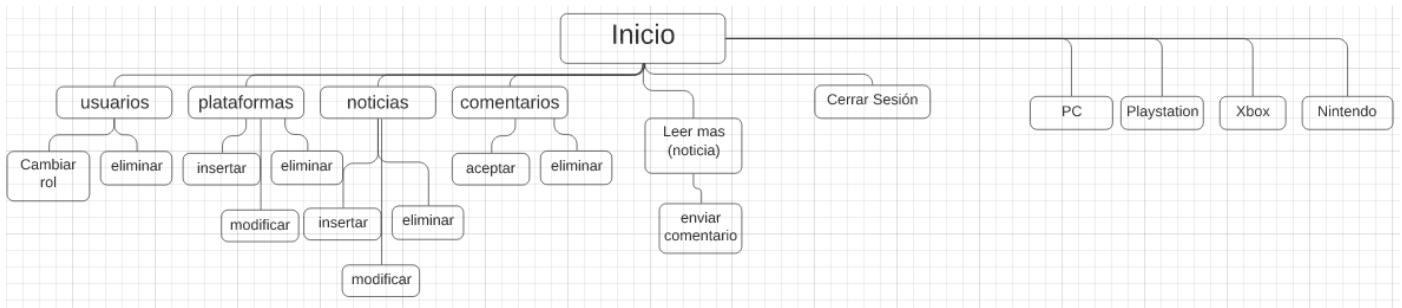
ANÁLISIS Y DISEÑO WEB

Existen dos secciones de la página:

- Zona Usuario y Usuario sin iniciar sesión: Los usuarios que no hayan iniciado sesión tendrán acceso a las publicaciones, pero no podrán verlas al completo ni comentar, al contrario, los usuarios que hayan iniciado sesión podrán comentar, ver la noticia al completo y podrán ver, modificar y eliminar su perfil.

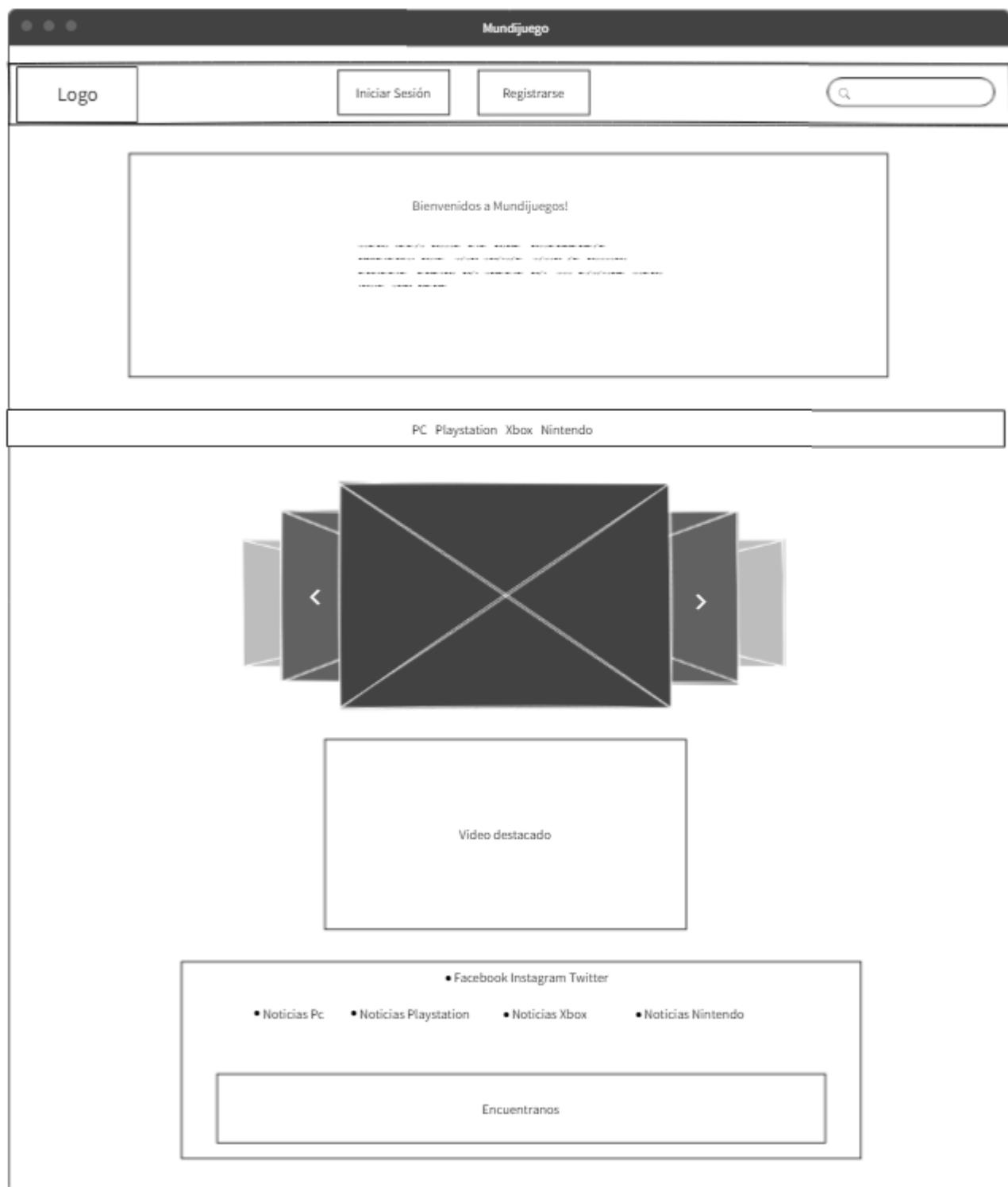


- **Zona Administrador:** Solo podrán acceder los usuarios con rol Admin, tienen control de todo el contenido de la página (noticias, plataformas, comentarios y usuarios), pueden crear, modificar y eliminar noticias, plataformas, cambiar el rol de un usuario y eliminarlo, y aceptar o rechazar un comentario.



Prototipo web y boceto de la estructura

Inicio



Login:

Mundijuego

Logo

Inicio

Registrarse

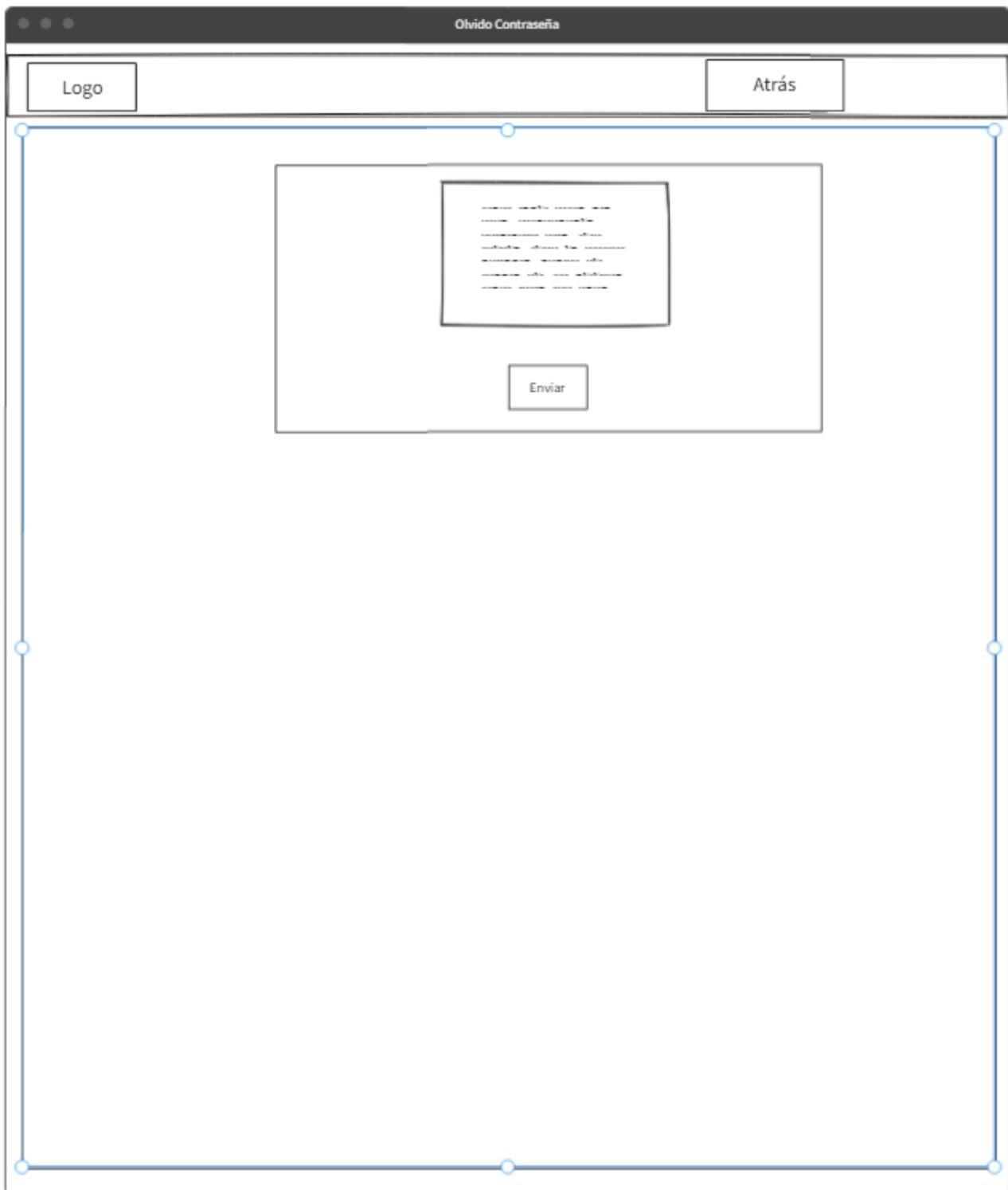
Usuario

Password

He olvidado mi contraseña

Enviar

Olvido Contraseña:



Registro:

The diagram illustrates a registration form interface. At the top, there is a header bar with three dots on the left, the text "Mundijuego" in the center, and two buttons, "Inicio" and "Iniciar sesión", on the right. Below the header is a large rectangular input area. Inside this area, the word "Usuario" is centered at the top. Following this are fifteen horizontal input fields, each preceded by a label: "Nombre", "Contraseña", "Primer Apellido", "Segundo Apellido", "Email", "DNI", "Direccion", "Fecha Nacimiento", "Telefono", "Codigo Postal", "Provincia", and "Municipio". At the bottom of the input area is a single "Enviar" button.

Logo

Mundijuego

Inicio

Iniciar sesión

Usuario

Nombre

Contraseña

Primer Apellido

Segundo Apellido

Email

DNI

Direccion

Fecha Nacimiento

Telefono

Codigo Postal

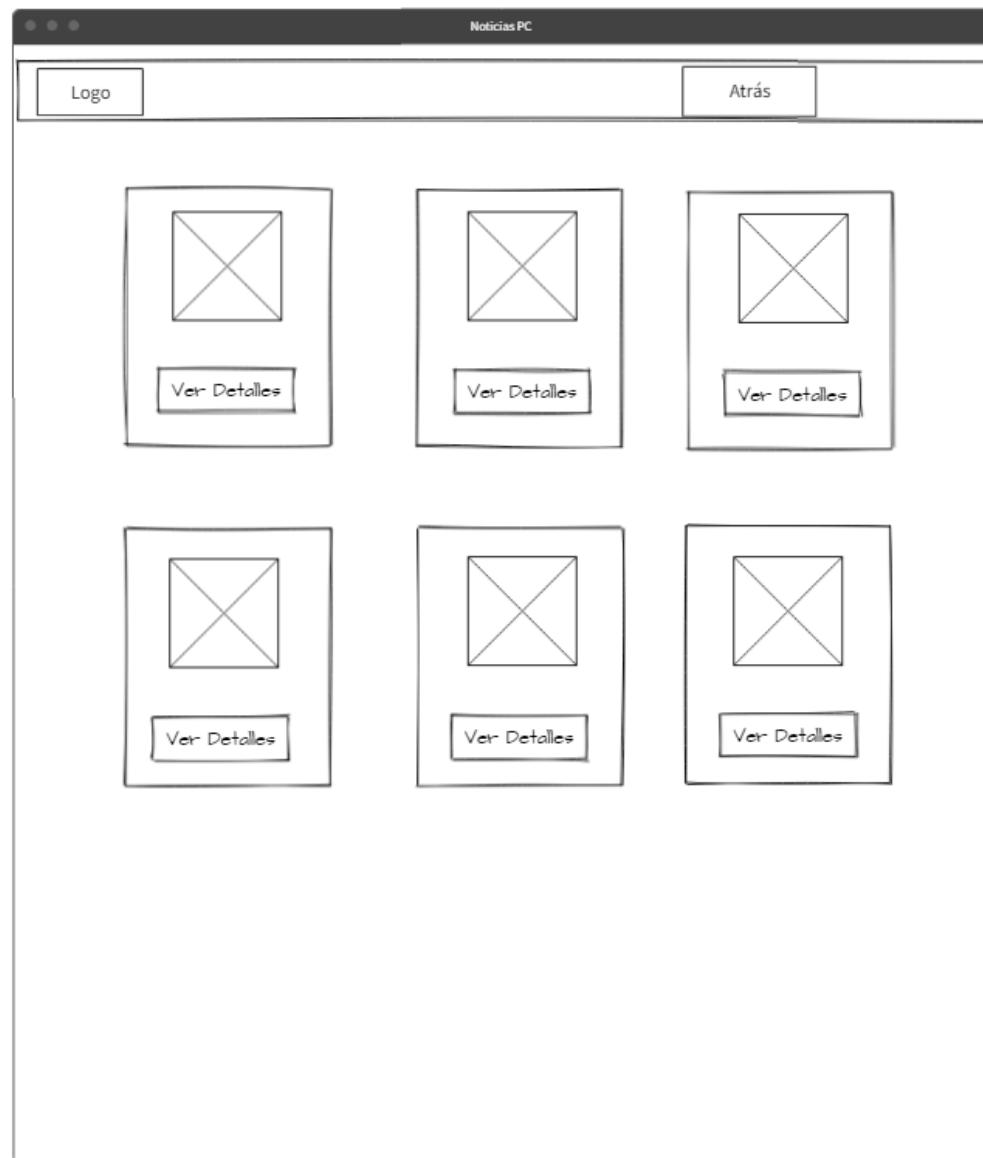
Provincia

Municipio

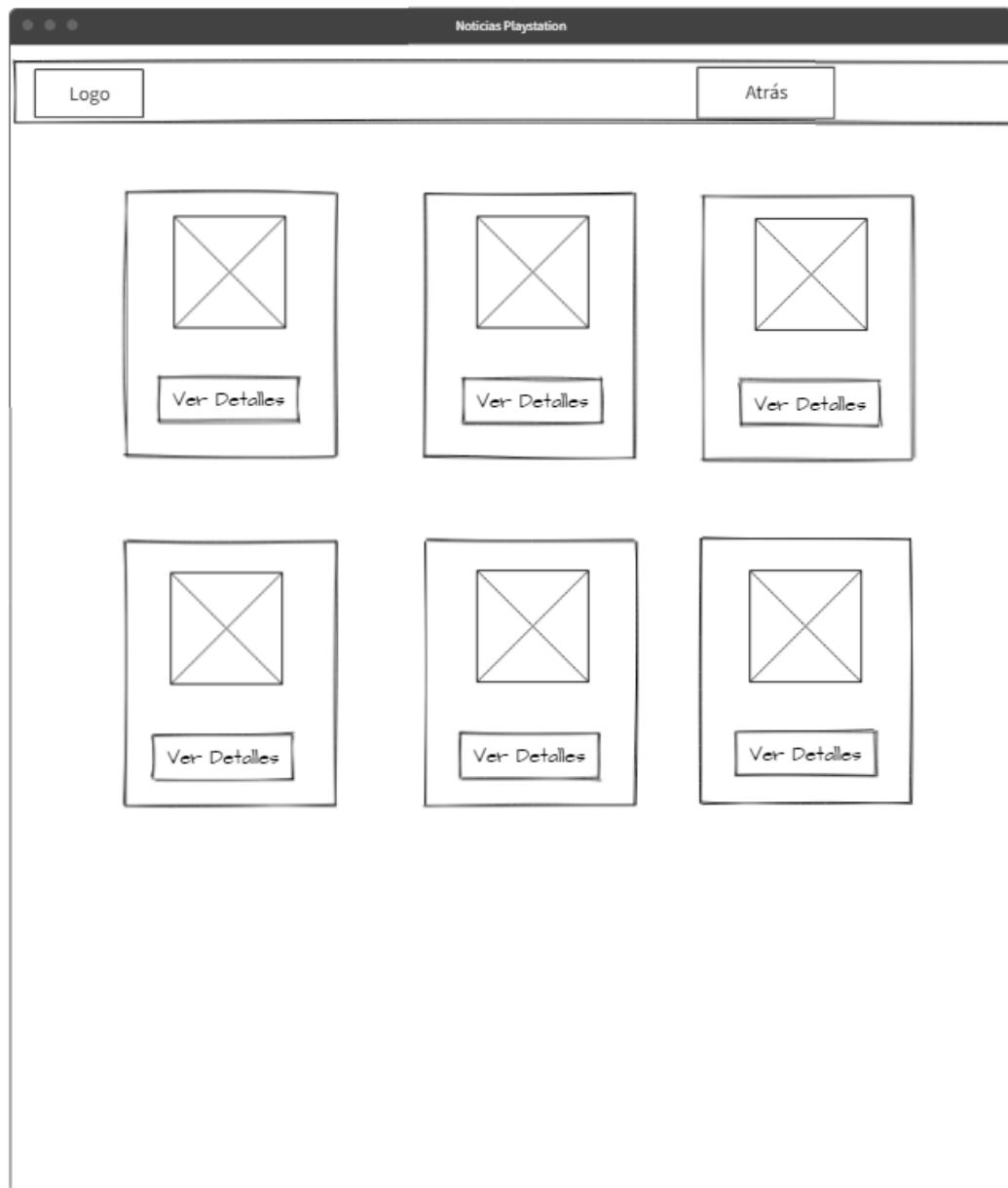
Enviar

Sección noticias videojuegos:

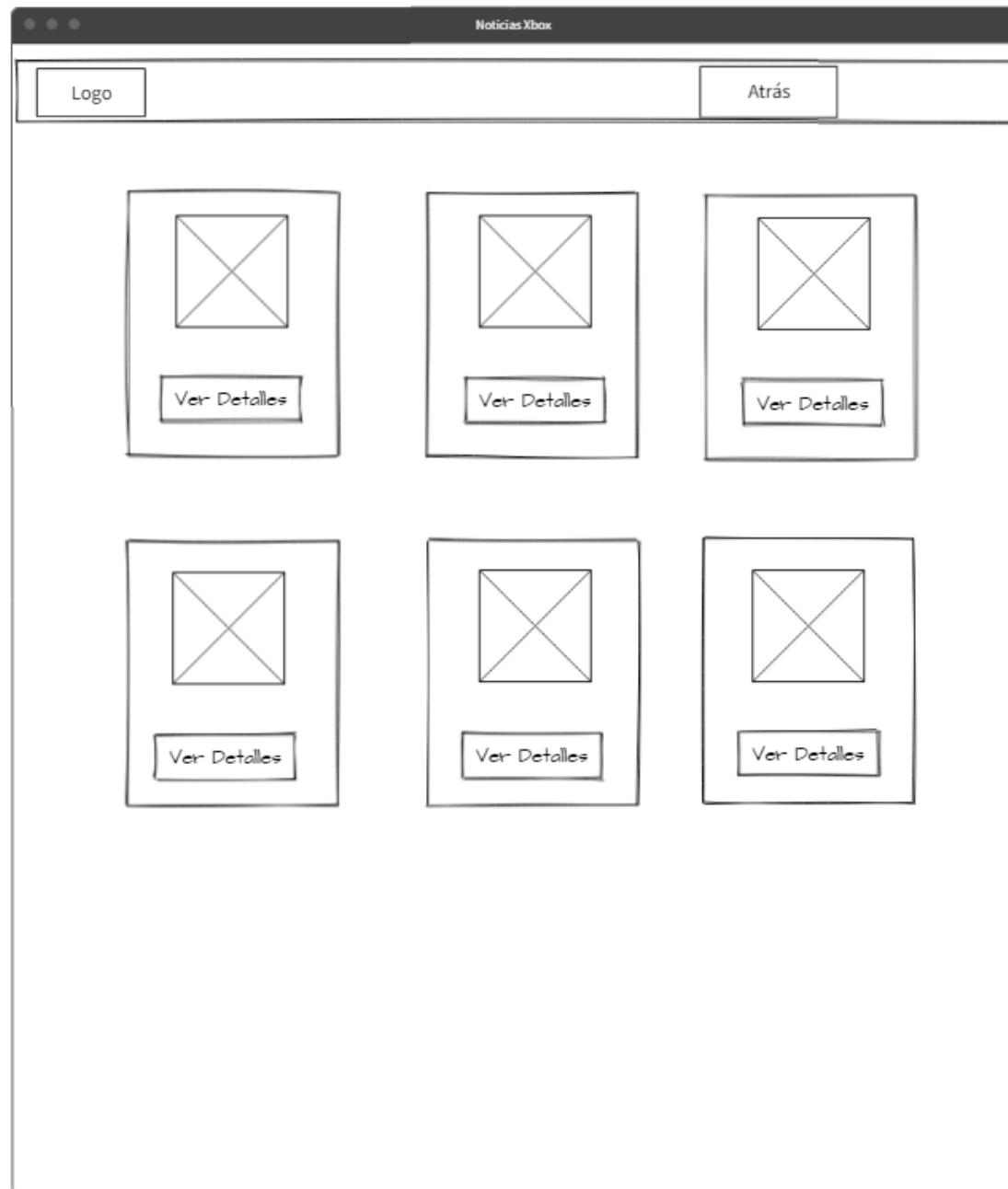
Noticias PC:



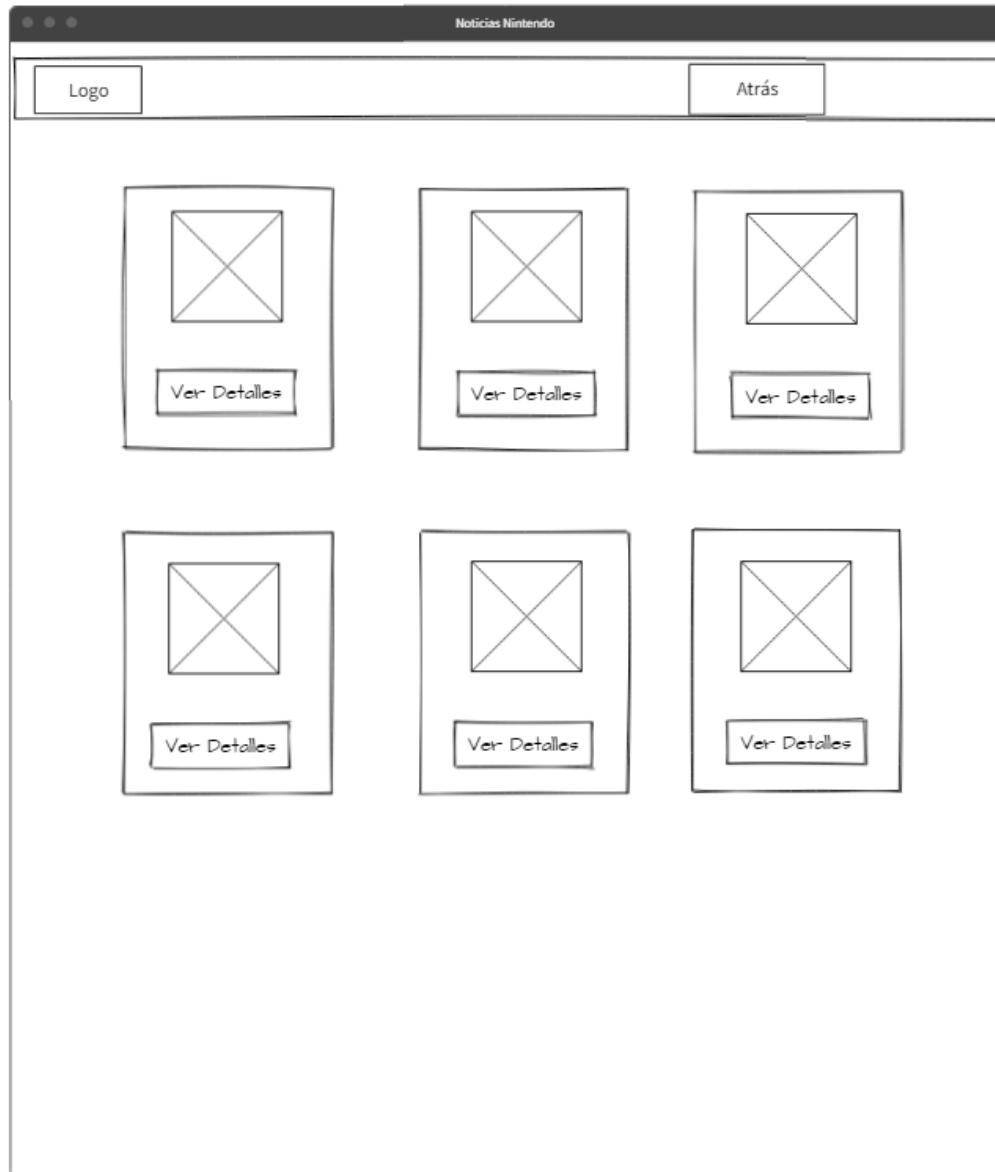
Noticias Playstation



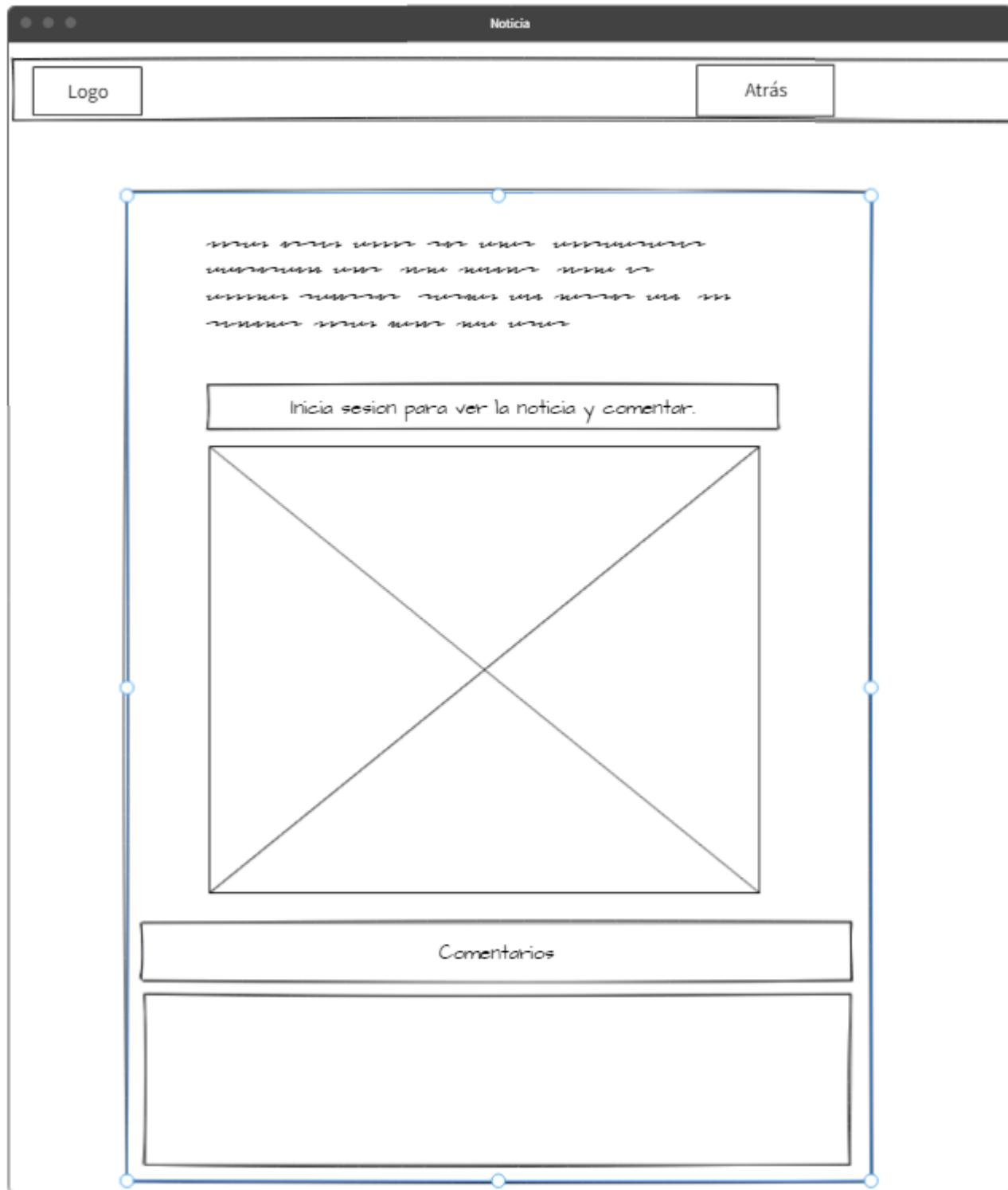
Noticias Xbox:



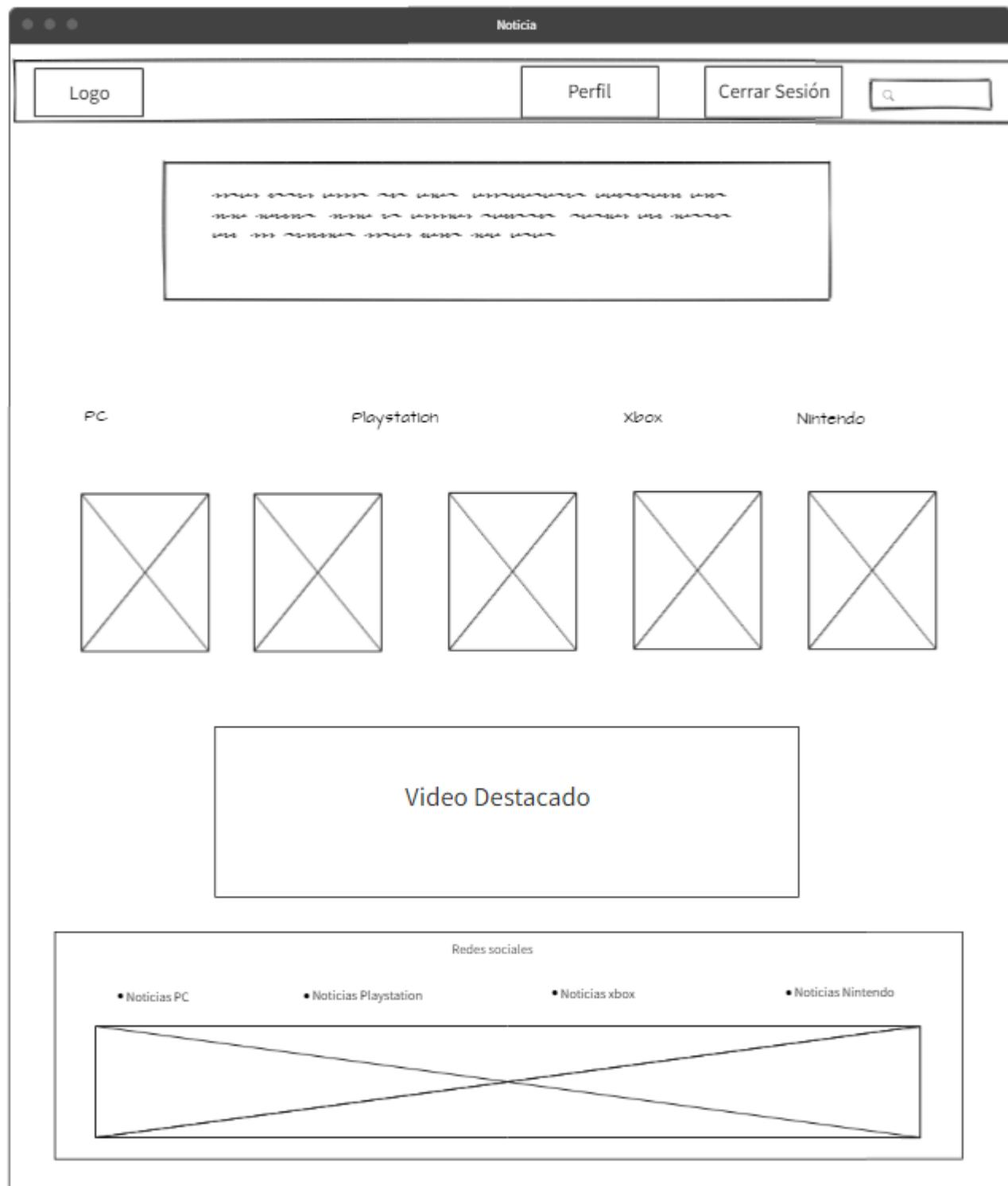
Noticias Nintendo:



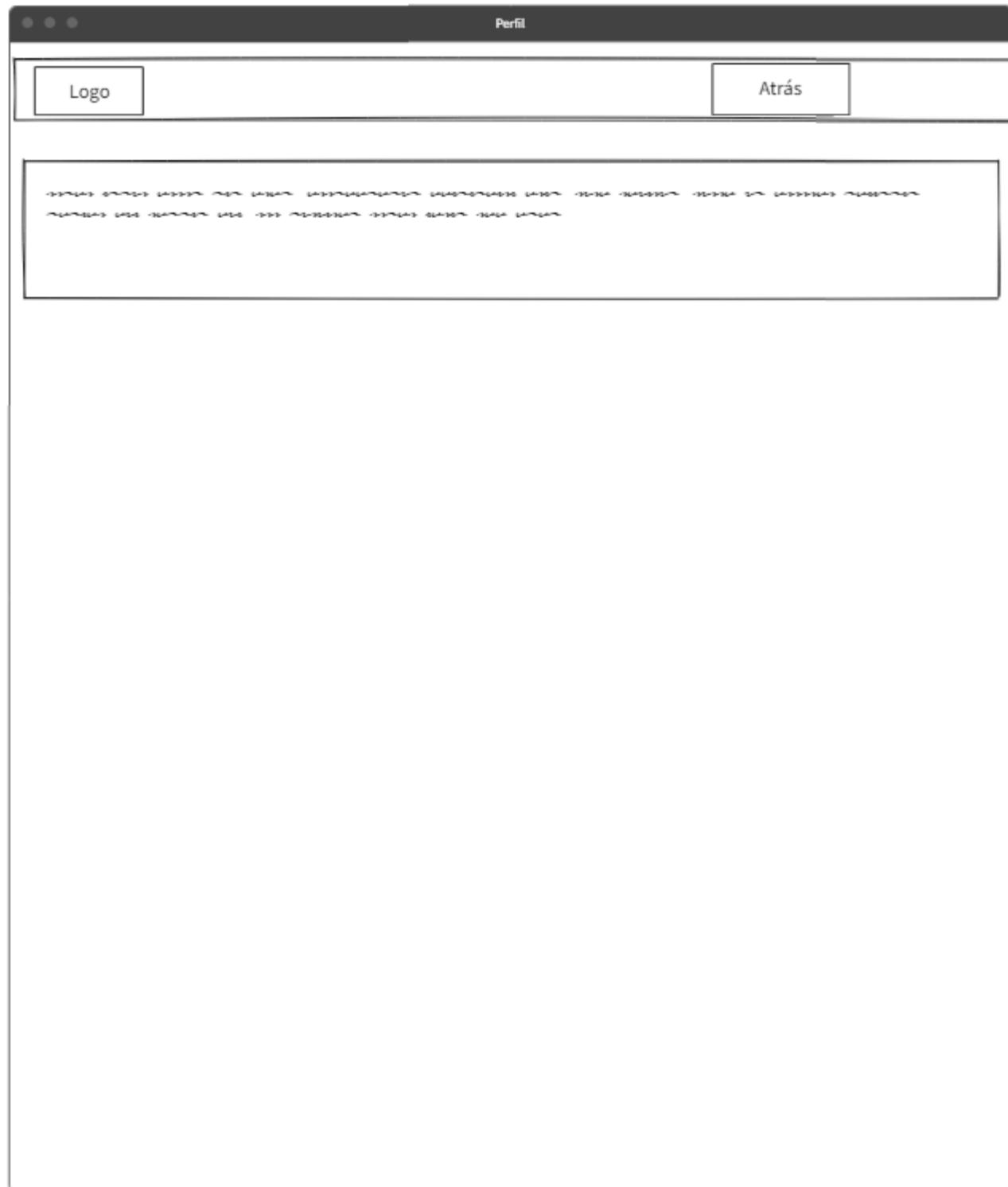
Detalles noticia cuando el usuario no ha iniciado sesión:



Página de inicio de un usuario normal:



Página del perfil de un usuario normal:



Modificar perfil usuario normal:

Modificar Perfil

Logo Atrás

Usuario

Password

Nombre

Primer Apellido

Segundo Apellido

Teléfono

Email

DNI

Dirección

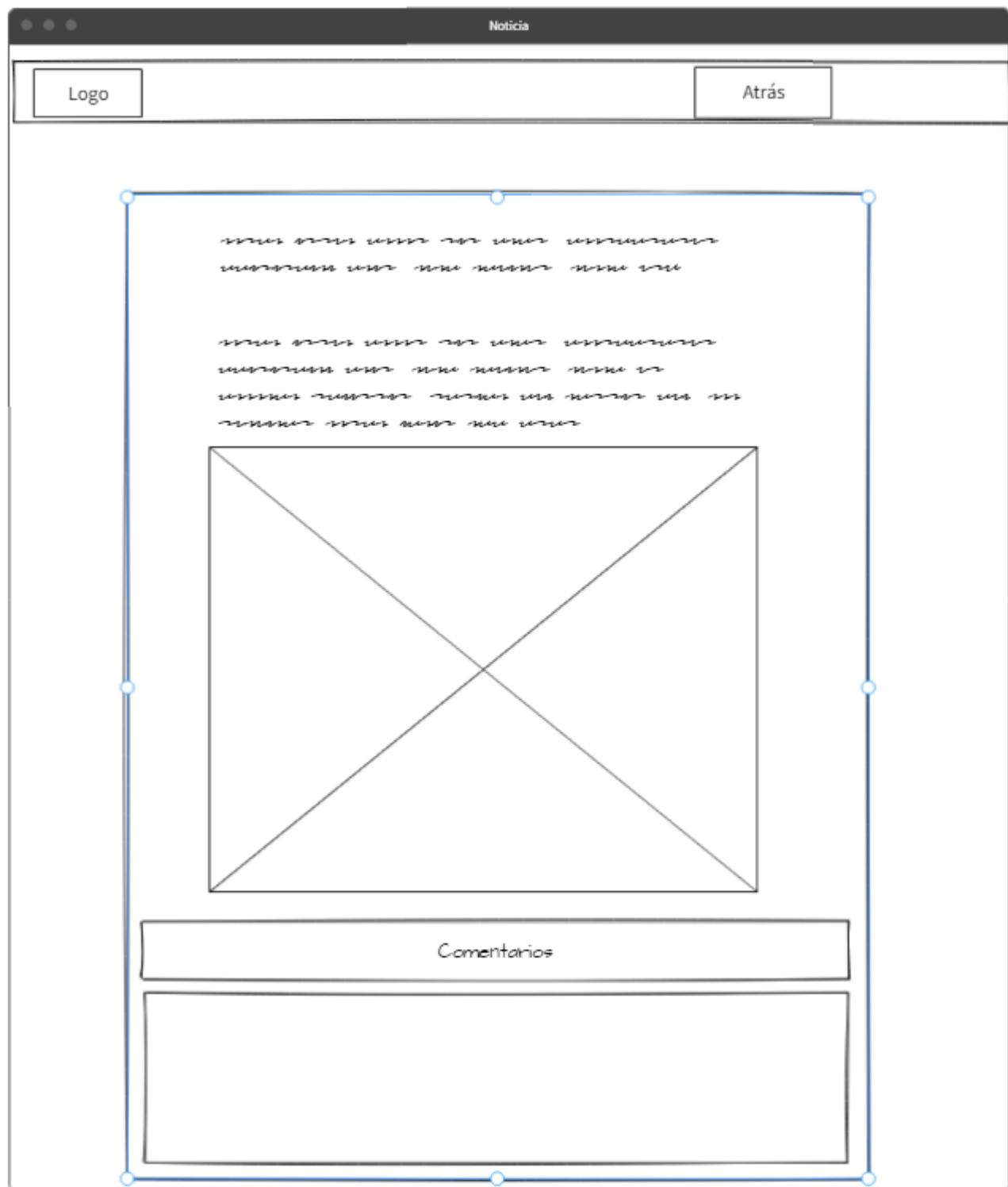
CP

Provincia

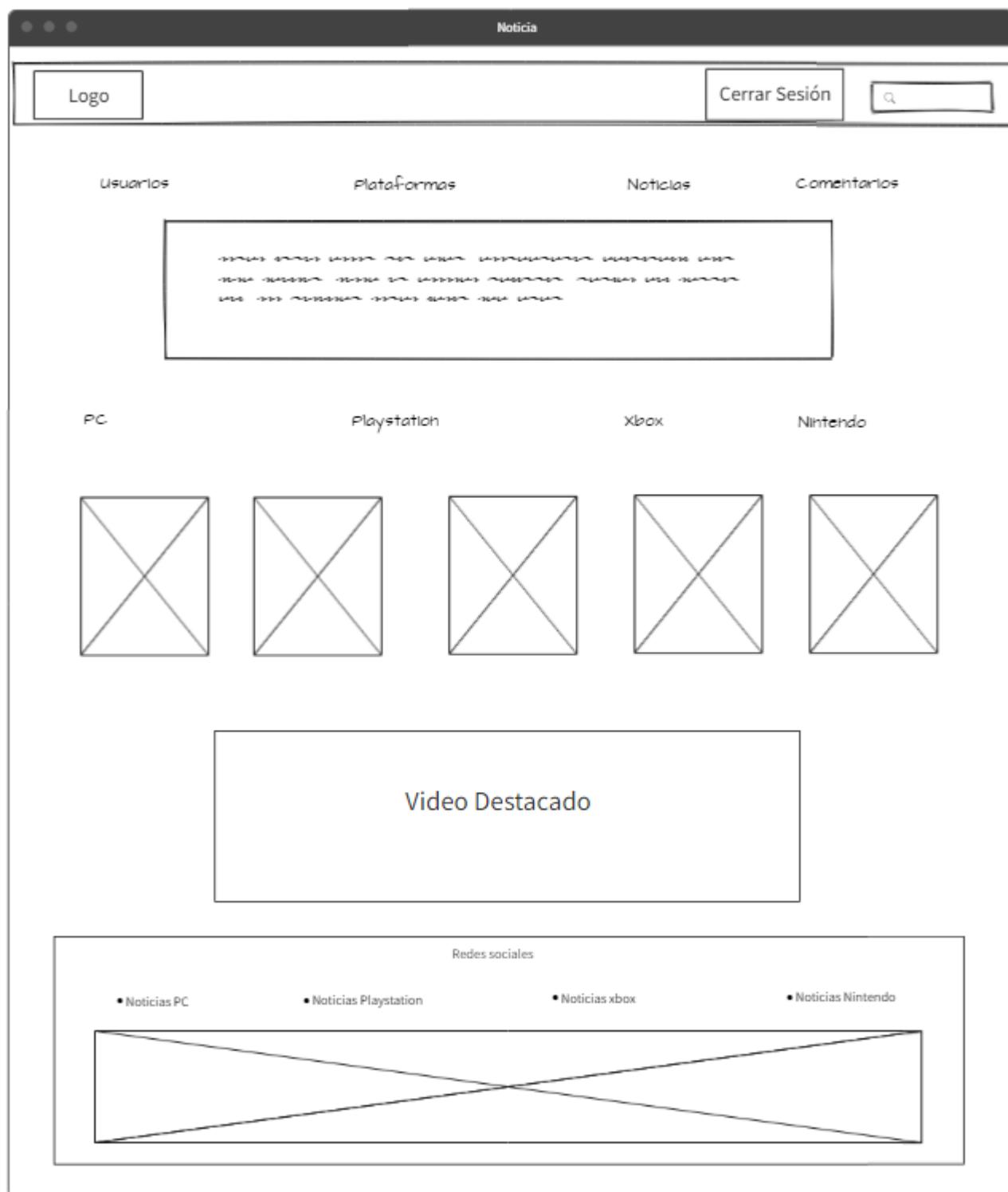
Comunidad Autónoma

Fecha Nacimiento

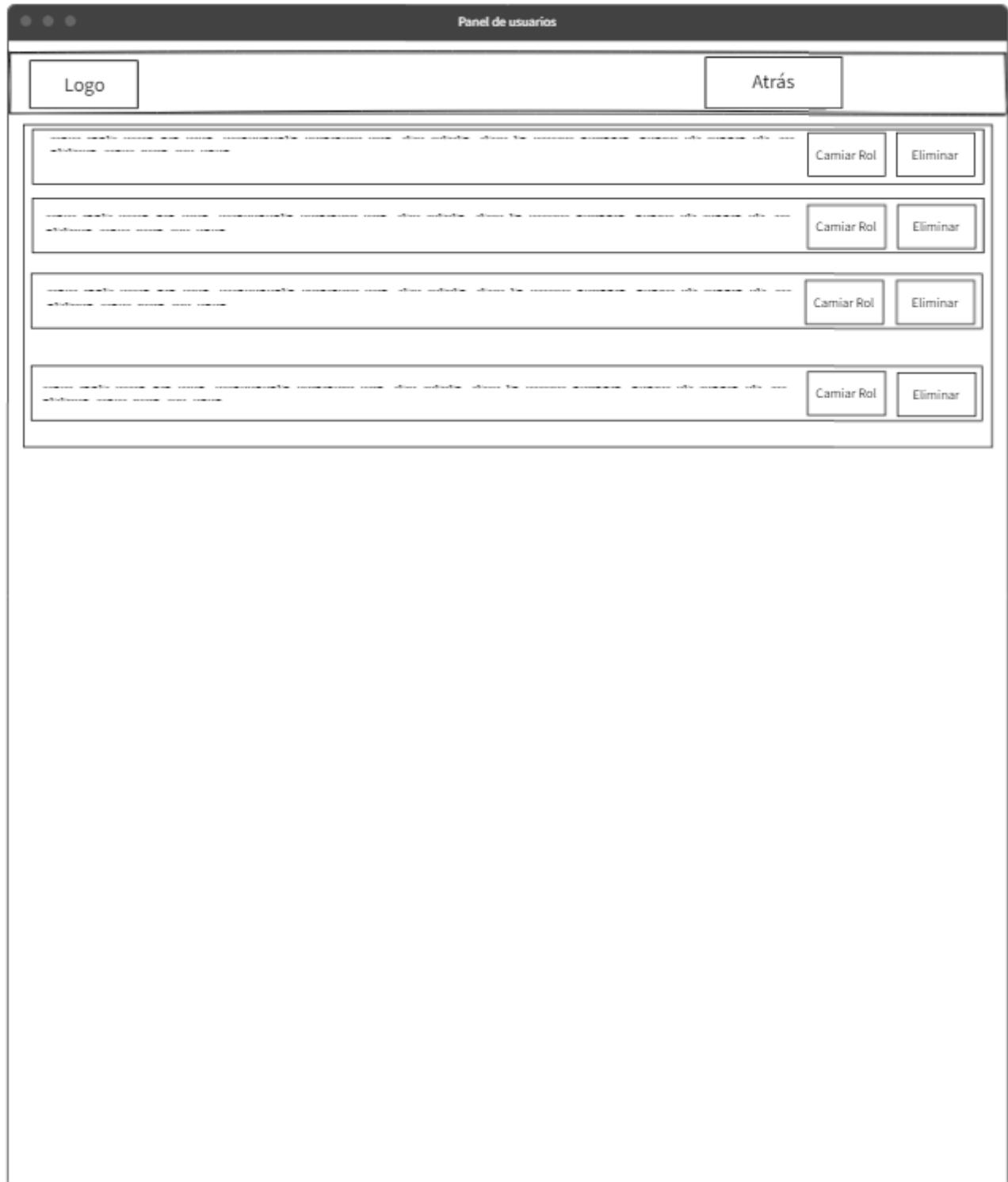
Detalles noticia cuando el usuario ha iniciado sesión:



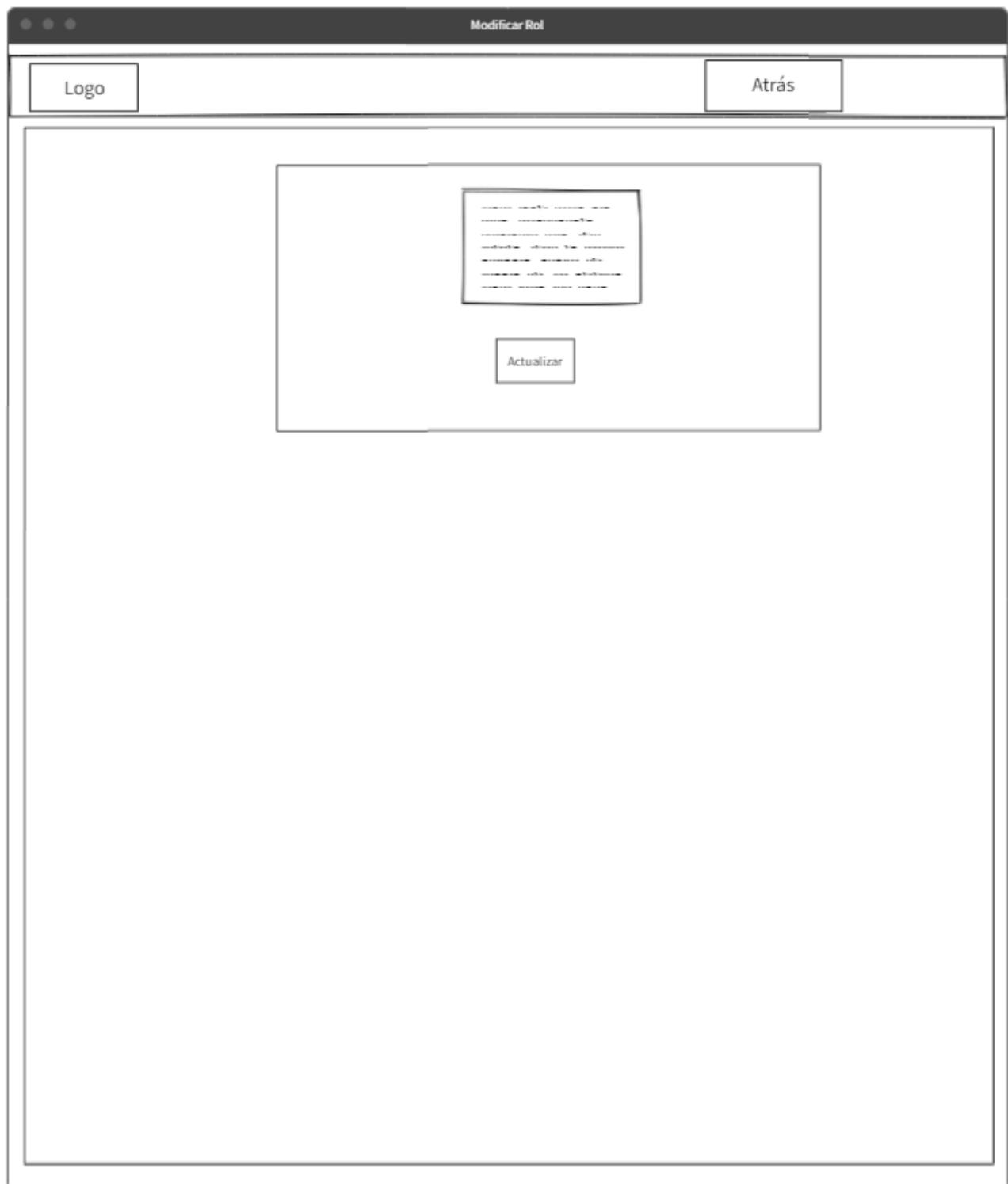
Página de inicio de un usuario Admin:



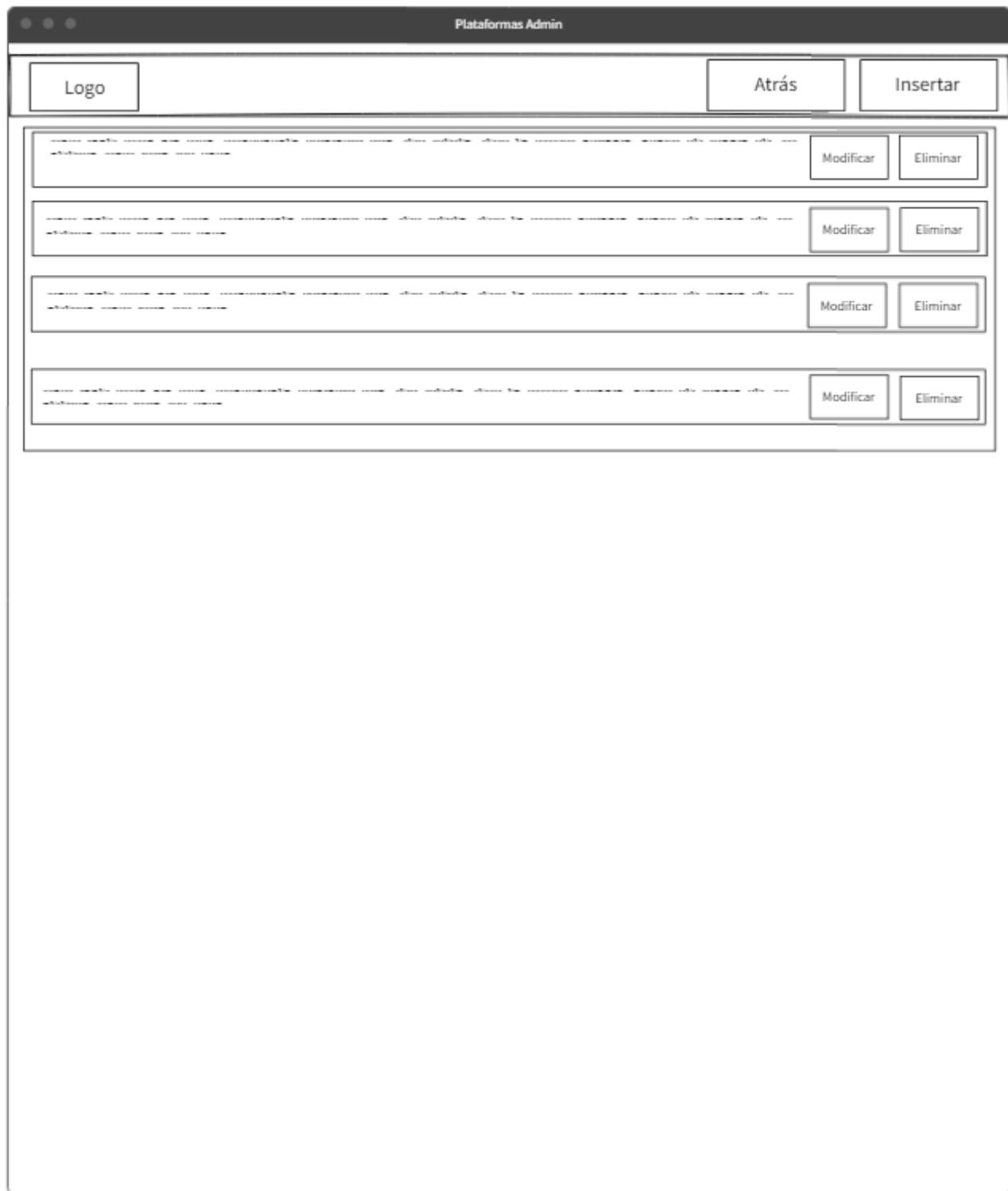
Sección Usuarios, administrador:



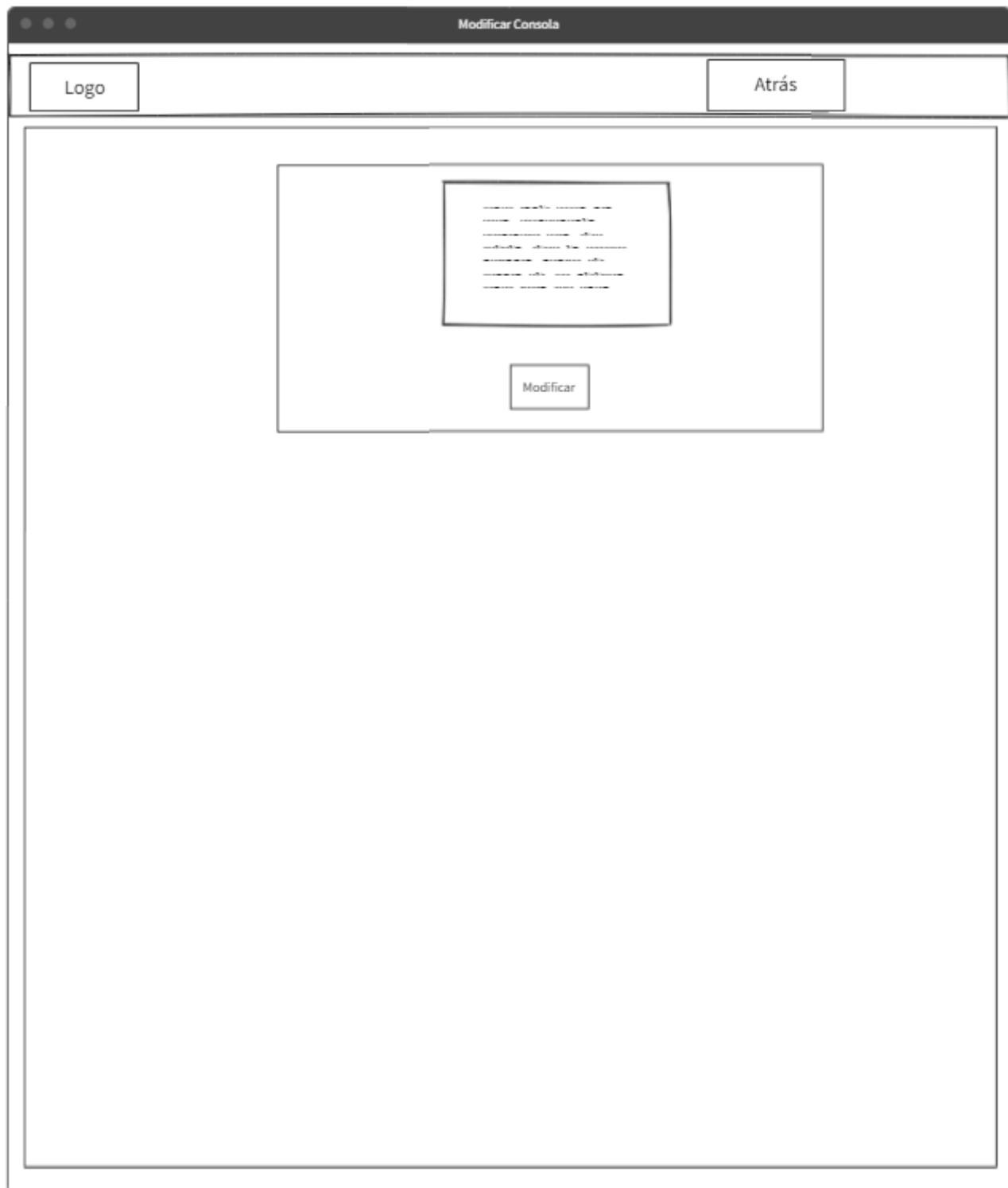
Sección Modificar Rol del usuario, administrador:



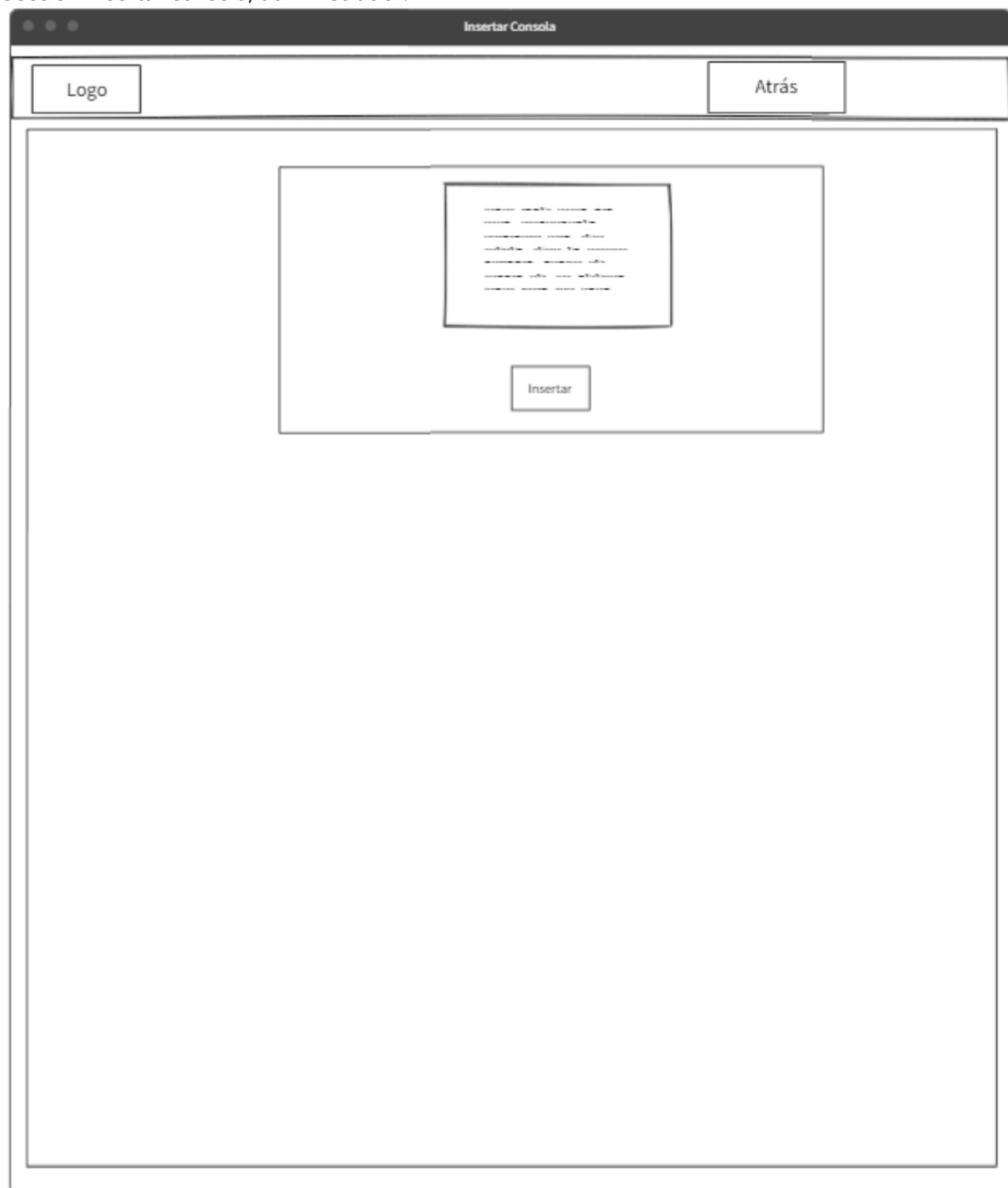
Sección Plataformas, administrador:



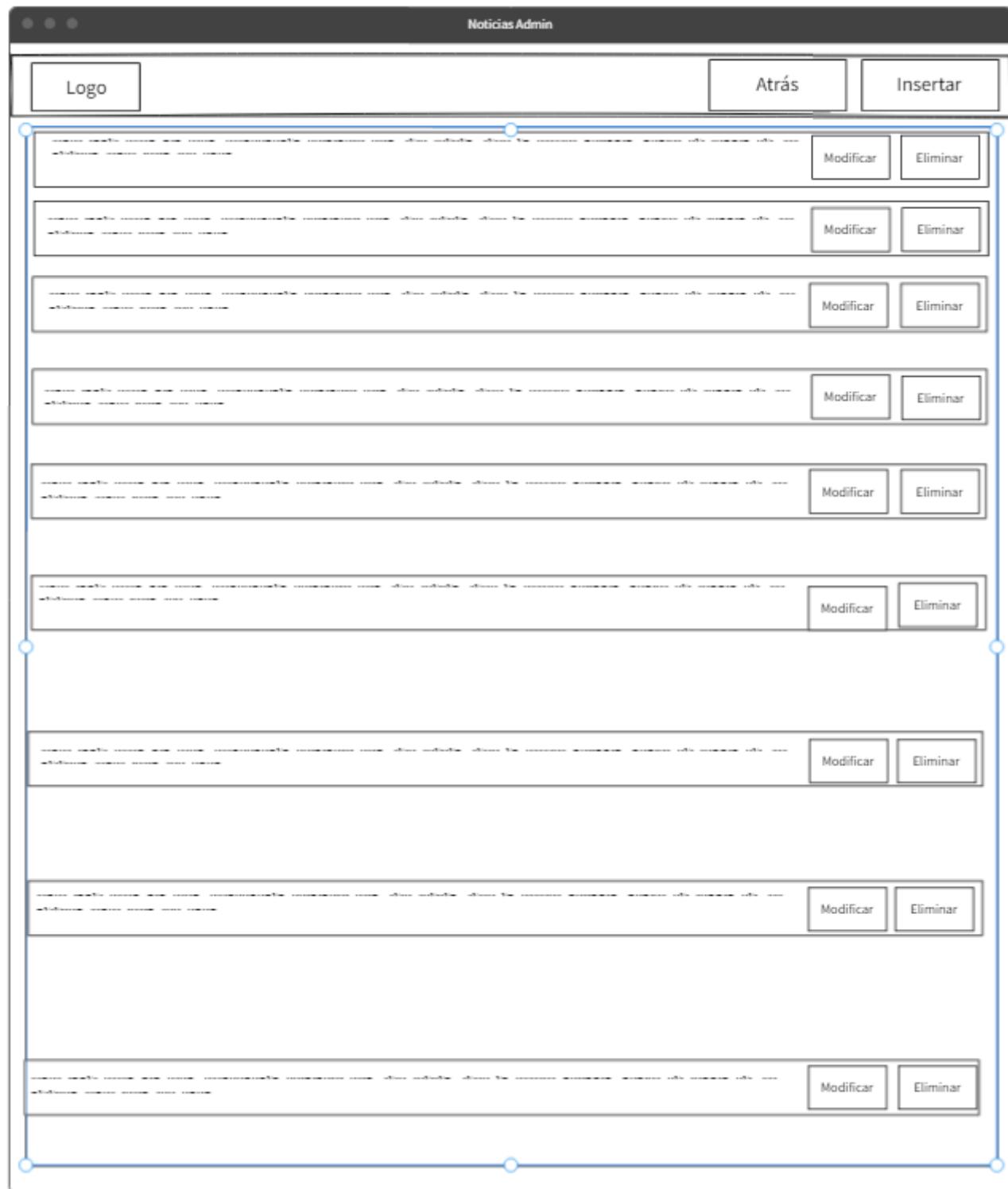
Sección Modificar plataforma, administrador:



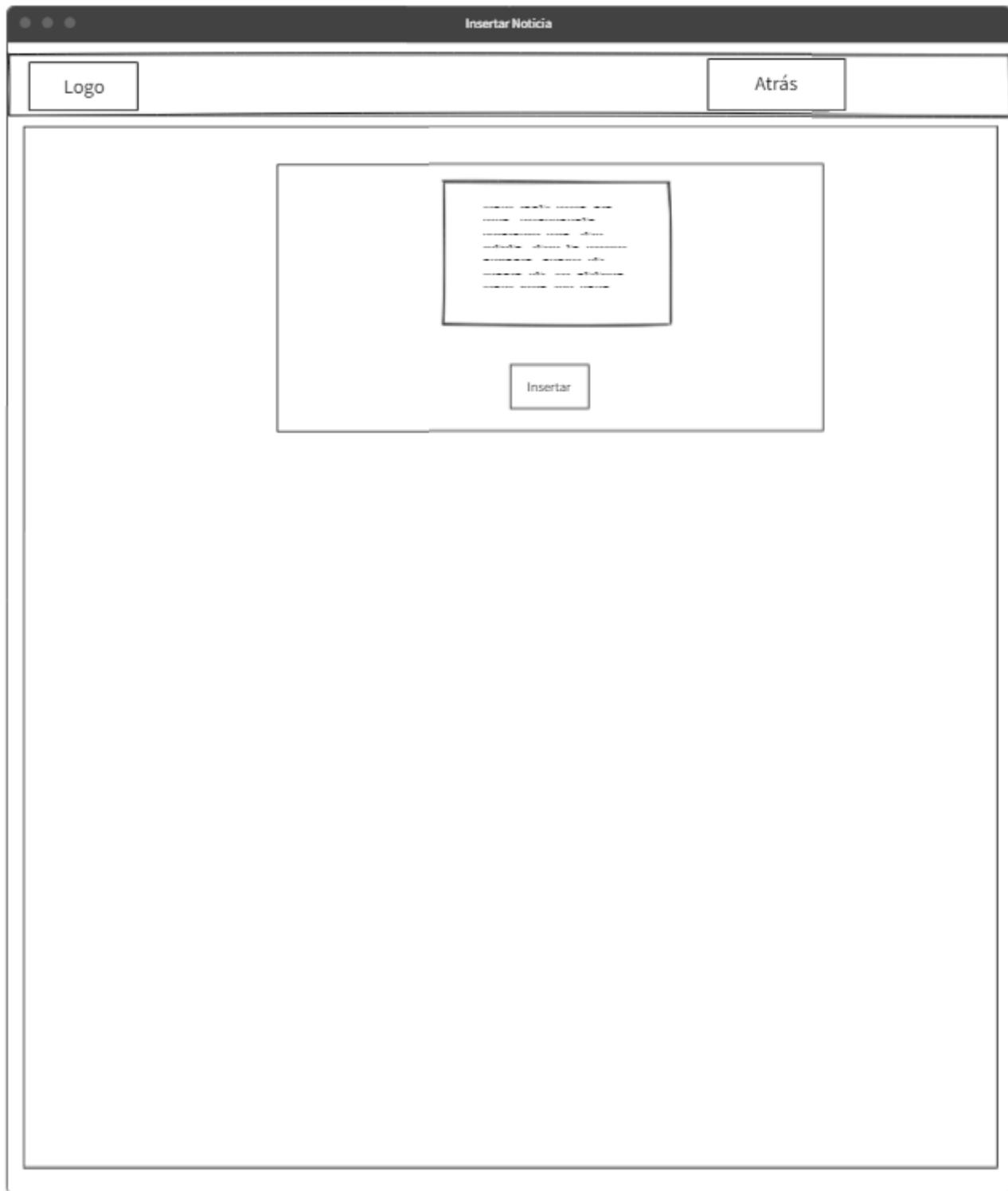
Sección insertar consola, administrador:



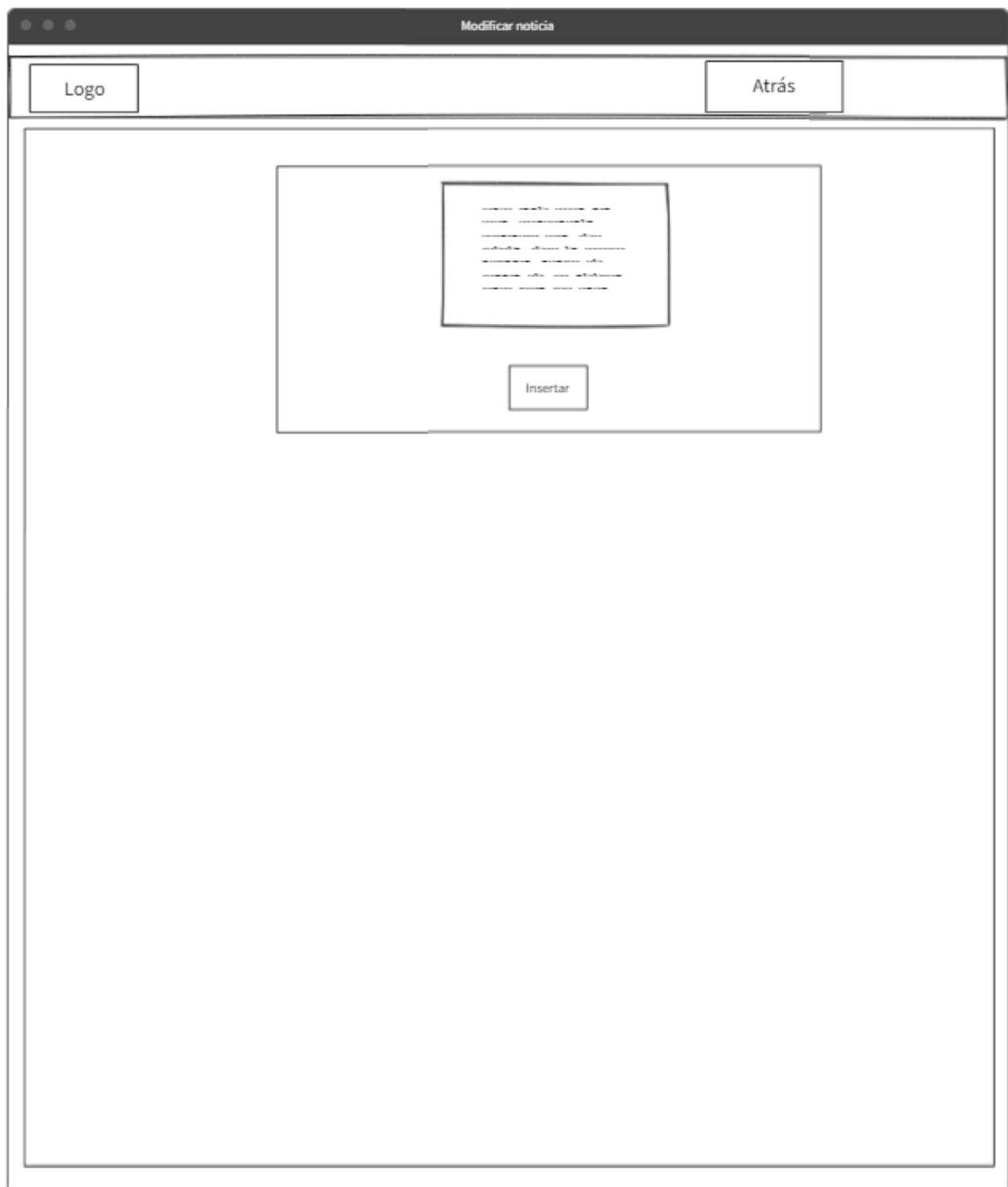
Sección Noticias, administrador:



Sección Insertar noticias, administrador:



Sección modificar noticias, administrador:



Sección comentarios, administrador:

ComprobarComentario

Logo	Atrás
	Aceptar Eliminar

Guía de estilos

Logo



Colores corporativos del logo

He utilizado este logo con estos colores que son los que más se usan en todas las páginas, me parecen unos colores simples y no llamativos.



Así también se elimina el texto y se queda solo el logo principal para ponerlo en las pestañas de la página.

Justificación de diseño del logo

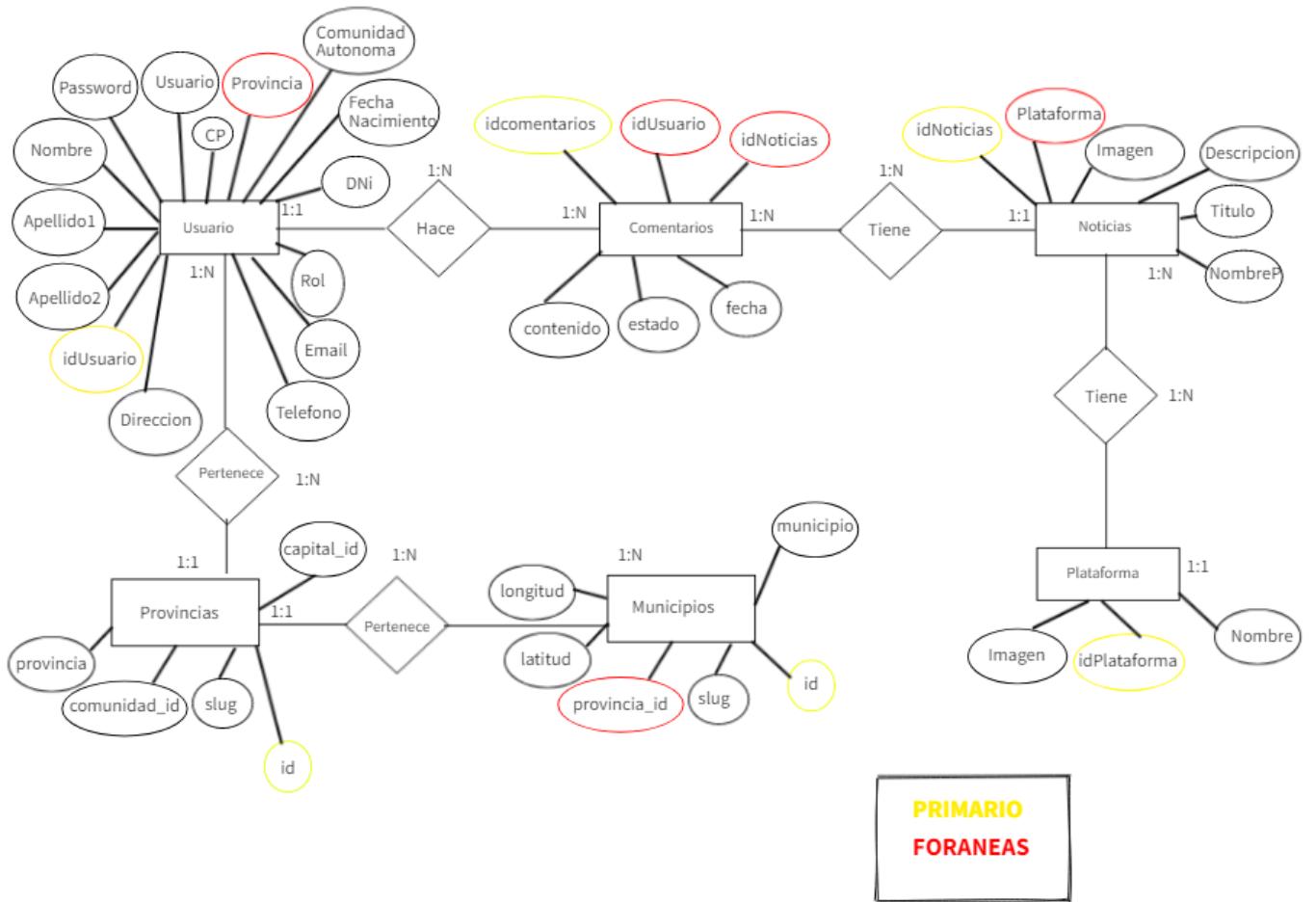
El mando en medio de la consola es ideal porque significa que la pagina está relacionada con los videojuegos, así como alrededor se muestra 'mundijuegos', este nombre lo engloba todo lo relacionado con los videojuegos.

Planificación de tareas:

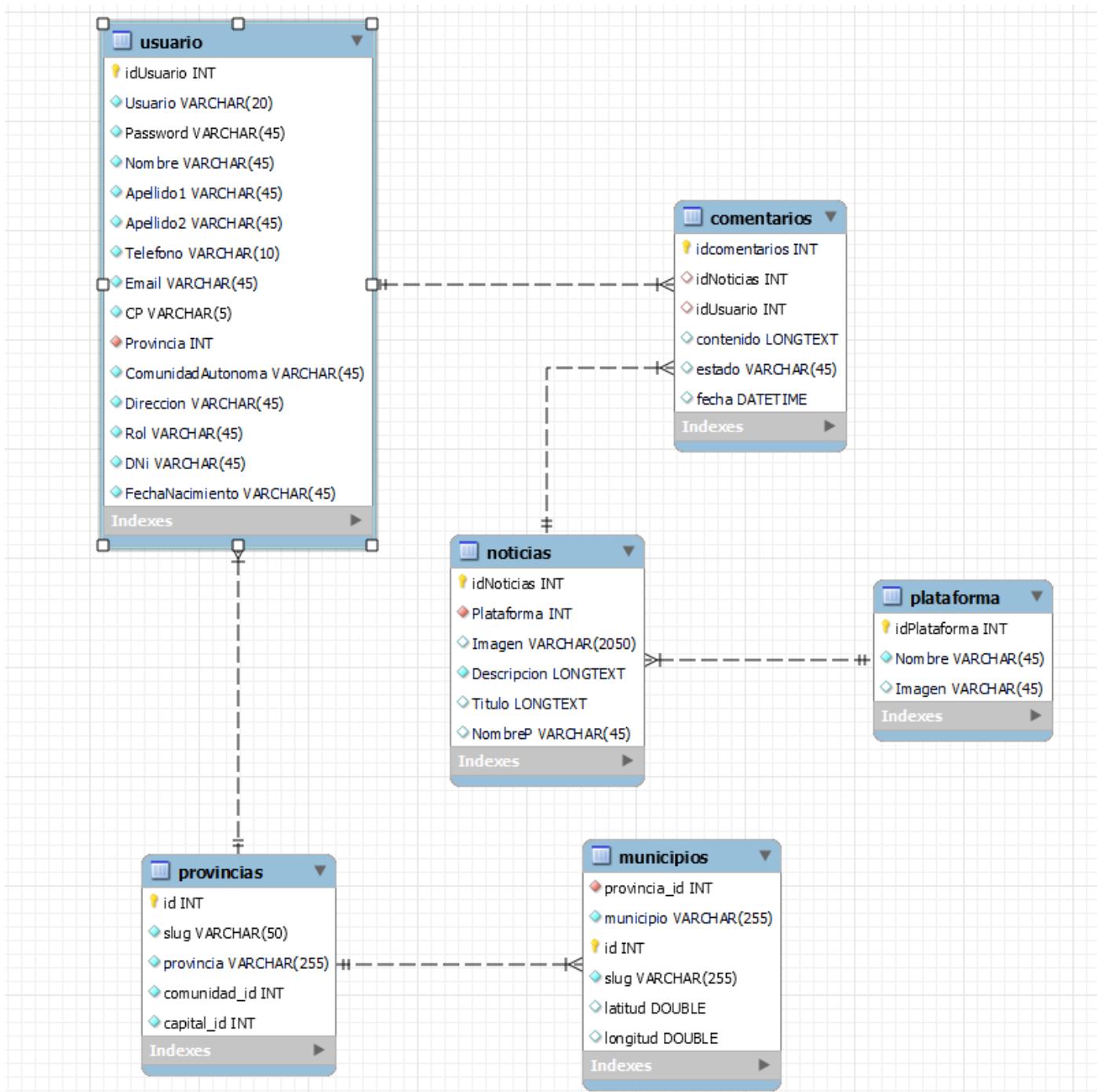


Base de datos

Diseño Entidad Relación de la BBDD



Modelo relacional BBDD



Script de creación BBDD:

- Tabla comentarios:

```
CREATE TABLE `comentarios` (
  `idcomentarios` int NOT NULL AUTO_INCREMENT,
  `idNoticias` int DEFAULT NULL,
  `idUsuario` int DEFAULT NULL,
  `contenido` longtext,
  `estado` varchar(45) DEFAULT NULL,
  `fecha` datetime DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`idcomentarios`),
  KEY `foranea noticia_idx` (`idNoticias`),
  KEY `foranea usuario_idx` (`idUsuario`),
  CONSTRAINT `foranea noticia` FOREIGN KEY (`idNoticias`) REFERENCES `noticias` (`idNoticias`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `foranea usuario` FOREIGN KEY (`idUsuario`) REFERENCES `usuario` (`idUsuario`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=61 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;
```

- Tabla municipios:

```
CREATE TABLE `municipios` (
  `provincia_id` int unsigned NOT NULL,
  `municipio` varchar(255) CHARACTER SET utf8 COLLATE utf8_spanish_ci NOT NULL,
  `id` int unsigned NOT NULL AUTO_INCREMENT,
  `slug` varchar(255) CHARACTER SET utf8 COLLATE utf8_spanish_ci NOT NULL,
  `latitud` double DEFAULT NULL,
  `longitud` double DEFAULT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `IDX_municipio` (`provincia_id`,`municipio`),
  UNIQUE KEY `slug` (`slug`),
  CONSTRAINT `provinciaMunicipio` FOREIGN KEY (`provincia_id`) REFERENCES `provincias` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=8117 DEFAULT CHARSET=utf8mb3 COLLATE=utf8_spanish_ci CHECKSUM=1
DELAY_KEY_WRITE=1 ROW_FORMAT=DYNAMIC;
/*!40101 SET character_set_client = @saved_cs_client */;
```

- Tabla noticias:

```
CREATE TABLE `noticias` (
  `idNoticias` int NOT NULL AUTO_INCREMENT,
  `Plataforma` int NOT NULL,
  `Imagen` varchar(2050) CHARACTER SET utf8 COLLATE utf8_bin DEFAULT NULL,
  `Descripcion` longtext CHARACTER SET utf8 COLLATE utf8_bin NOT NULL,
  `Titulo` longtext CHARACTER SET utf8 COLLATE utf8_bin,
  `NombreP` varchar(45) CHARACTER SET utf8 COLLATE utf8_bin DEFAULT NULL,
  PRIMARY KEY (`idNoticias`),
  UNIQUE KEY `idNoticias_UNIQUE` (`idNoticias`),
  KEY `Plataforma_idx` (`Plataforma`),
  CONSTRAINT `idNoticias` FOREIGN KEY (`Plataforma`) REFERENCES `plataforma` (`idPlataforma`)
) ENGINE=InnoDB AUTO_INCREMENT=35 DEFAULT CHARSET=utf8mb3 COLLATE=utf8_bin;
/*!40101 SET character_set_client = @saved_cs_client */;
```

- Tabla plataformas:

```
CREATE TABLE `plataforma` (
  `idPlataforma` int NOT NULL AUTO_INCREMENT,
  `Nombre` varchar(45) NOT NULL,
  `Imagen` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`idPlataforma`),
  UNIQUE KEY `Nombre_UNIQUE` (`Nombre`),
  UNIQUE KEY `idPlataforma_UNIQUE` (`idPlataforma`)
) ENGINE=InnoDB AUTO_INCREMENT=38 DEFAULT CHARSET=utf8mb3;
/*!40101 SET character_set_client = @saved_cs_client */;
```

- Tabla provincias:

```
CREATE TABLE `provincias` (
  `id` int unsigned NOT NULL AUTO_INCREMENT,
  `slug` varchar(50) CHARACTER SET utf8 COLLATE utf8_spanish_ci NOT NULL,
  `provincia` varchar(255) CHARACTER SET utf8 COLLATE utf8_spanish_ci NOT NULL,
  `comunidad_id` int unsigned NOT NULL,
  `capital_id` int NOT NULL DEFAULT '-1',
  PRIMARY KEY (`id`),
  UNIQUE KEY `IDX_provincia` (`provincia`),
  KEY `FK_provincias` (`comunidad_id`)
) ENGINE=InnoDB AUTO_INCREMENT=53 DEFAULT CHARSET=utf8mb3 COLLATE=utf8_spanish_ci CHECKSUM=1
DELAY_KEY_WRITE=1 ROW_FORMAT=DYNAMIC;
/*!40101 SET character_set_client = @saved_cs_client */;
```

- Tabla usuarios:

```
CREATE TABLE `usuario` (
  `idUsuario` int NOT NULL AUTO_INCREMENT,
  `Usuario` varchar(20) NOT NULL COMMENT 'El usuario debe ser único en la Base de Datos. Además debe tener una longitud mínima de 6 caracteres.',
  `Password` varchar(45) NOT NULL COMMENT 'La contraseña no puede ser nula.\nTiene que tener una longitud mínima de 8 caracteres, contener números y algún carácter especial de la siguiente lista (!,@,#,$,%,&,?,¿,*€),',
  `Nombre` varchar(45) NOT NULL,
  `Apellido1` varchar(45) NOT NULL,
  `Apellido2` varchar(45) NOT NULL,
  `Telefono` varchar(10) NOT NULL COMMENT 'Se comprobará que solo puedan ser teléfonos nacionales',
  `Email` varchar(45) NOT NULL COMMENT 'Se comprobará que es un email válido',
  `CP` varchar(5) NOT NULL,
  `Provincia` int unsigned NOT NULL COMMENT 'Se comprobará que la provincia introducida concuerda con el CP introducido',
  `ComunidadAutonoma` varchar(45) NOT NULL COMMENT 'Se comprobará que la provincia seleccionada pertenece a la comunidad autónoma correspondiente',
  `Direccion` varchar(45) NOT NULL,
  `Rol` varchar(45) NOT NULL,
  `DNI` varchar(45) NOT NULL,
  `FechaNacimiento` varchar(45) NOT NULL,
  PRIMARY KEY (`idUsuario`),
  UNIQUE KEY `idUsuario_UNIQUE` (`idUsuario`),
  UNIQUE KEY `Usuario_UNIQUE` (`Usuario`),
  KEY `ProvinciasUsuario_idx` (`Provincia`),
  CONSTRAINT `ProvinciasUsuario` FOREIGN KEY (`Provincia`) REFERENCES `provincias` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=78 DEFAULT CHARSET=utf8mb3;
/*!40101 SET character_set_client = @saved_cs_client */;
```

Consultas

Las consultas a continuación se utilizan para todas las tablas del proyecto.

Ejemplos:

Seleccionar:

Si queremos mostrar todos los datos de los usuarios que tenemos registrados usaremos:

```
SELECT * FROM 'plataformas';
```

Si necesitamos solo unos datos o mostrar datos de un usuario en concreto utilizamos una condición:

```
SELECT * FROM usuarios WHERE ('Email' = '$email');
```

Insertar:

Se utiliza `INSERT INTO` para insertar datos a la BD.

Modificar:

Para modificar datos se utiliza `UPDATE **tabla** SET **datos**` y también mediante su ID.

Eliminar:

Para eliminar datos de la BD se utiliza `DELETE FROM` junto al ID que queremos eliminar.

Validación de formularios

Para validar los formularios del proyecto utilizaremos Javascript. Se ha utilizado el mismo funcionamiento del script para (login, registro, recuperar contraseña y contraseña nueva) pero separados por archivos. Cada archivo tiene scripts y expresiones regulares para evitar escribir contenido no permitido en ciertos campos.

Formularios:

Login:

Registro:

Registro x + - _

localhost/teste/ingresar_usuario.php

 MUNDIJUEGOS

Inicio Inicia sesión

Usuario:
Alumno98

Formato Incorrecto
Nombre:
Nombre

Formato Incorrecto
Contraseña:
La contraseña debe tener un mínimo de 8 caracteres, contener una mayúscula y un carácter especial.

Primer Apellido:
1º apellido

Formato Incorrecto
Segundo Apellido:
2º apellido

Formato Incorrecto
Email:
Email

Formato Incorrecto
DNI:
DNI

Formato Incorrecto
Direccion:
Calle....

Imagen del funcionamiento del script:

Registro

localhost/este/ingresar_usuario.php

Segundo Apellido
2º apellido

Formato Incorrecto
Email:
Email

Formato Incorrecto
DNI:
DNI

Formato Incorrecto
Direccion:
Calle:...

Fecha de nacimiento:
dd/mm/aaaa

Telefono:
pp

Codigo Postal:
Ejemplo: 52005

Formato Incorrecto
Provincia: Elija una provincia

Municipio:
 Elija una opcion...

Enviar

Olvido de la contraseña:

Olvido Contraseña

localhost/este/contraseña_nueva.html

MUNDIJUEGOS

Atrás

Usuario
 Alumno98

El formato es incorrecto
Email
 Email

El formato es incorrecto
Enviar

Recuperar contraseña:

Recuperar contraseña

localhost/este/recuperar_contraseña.html

MUNDIJUEGOS

Atrás

Contraseña nueva:

Enviar

Funcionamiento del script:

Login:

```
//constantes que almacenan los identificadores
const usuario = document.getElementById("usuario")
const contrasena = document.getElementById("contrasena");
const boton = document.getElementById("boton");
//oculta los mensajes de errores
document.getElementById("usuario_error").style.visibility = "hidden";
document.getElementById("contrasena_error").style.visibility = "hidden";

//constantes para las expresiones regulares
const expresiones = {
    usuario: /^[a-zA-Z0-9À-Ñ_]{6,}$/,
    contrasena: /^(?=.{8,}).*[a-z](?=.*[A-Z])(?=.*[0-9])(?=.*\W).*$/
}
//funcion para validar los usuarios
function validarUsuario(){
    if (expresiones.usuario.test(usuario.value)) {
        document.getElementById("usuario_error").style.visibility = "hidden";
    }else{
        document.getElementById("usuario_error").style.visibility = "visible";
        document.getElementById("usuario_error").style.color = "red";
    }
}

//funcion para validar la contraseña
function validarContrasena(){
    if (expresiones.contrasena.test(contrasena.value)) {
        document.getElementById("contrasena_error").style.visibility = "hidden";
    }else{
        document.getElementById("contrasena_error").style.visibility = "visible";
        document.getElementById("contrasena_error").style.color = "red";
    }
}

//funcion para el boton
function pulsarBoton(){
    validarUsuario();
    validarContrasena();
}

//oyentes de eventos
usuario.addEventListener("keyup", validarUsuario);
usuario.addEventListener("blur", validarUsuario);
contra.addEventListener("keyup", validarContrasena);
contra.addEventListener("blur", validarContrasena);
boton.addEventListener("click", pulsarBoton);
```

Registro:

```

//constantes que almacenan los identificadores del html
const usuario = document.getElementById("usuario");
const email = document.getElementById("email");
const contrasena = document.getElementById("contrasena");
const nombre = document.getElementById("nombre");
const apellido1 = document.getElementById("apellido1");
const dni = document.getElementById("dni");
const boton = document.getElementById("boton");
const telefono = document.getElementById('telefono');
const codigopostal = document.getElementById('codigopostal');
const provincia = document.getElementById('provincia');
const municipios = document.getElementById('municipios');
const apellido2 = document.getElementById("apellido2");
const direccion = document.getElementById("direccion");
const fechanacimiento = document.getElementById("fechanacimiento");
const rol = document.getElementById("rol");
const errorUsuario=document.getElementById("usuario_error");

//ocultamos los mensajes de errores con JQUERY
$("#usuario_error").css("visibility", "hidden");
$("#contrasena_error").css("visibility", "hidden");
$("#dni_error").css("visibility", "hidden");
$("#apellido1_error").css("visibility", "hidden");
$("#nombre_error").css("visibility", "hidden");
$("#codigopostal_error").css("visibility", "hidden");
$("#provincia_error").css("visibility", "hidden");
$("#municipio_error").css("visibility", "hidden");
$("#telefono_error").css("visibility", "hidden");
$("#apellido2_error").css("visibility", "hidden");
$("#email_error").css("visibility", "hidden");
$("#direccion_error").css("visibility", "hidden");
$("#fechanacimiento_error").css("visibility", "hidden");
$("#rol_error").css("visibility", "hidden");
//constante para las expresiones regulares
const expresiones = {
    usuario: /^[a-zA-Z0-9_\-\ ]{6,}$/,
    email: /^[a-zA-Z0-9\.\+\-]+@[a-zA-Z0-9\.\-]+\.[a-zA-Z0-9\.\-\.]+\$/,
    dni: /^[0-9]{8}[A-Za-z]$/,
    contrasena: /^(?=.*[A-Z])(?=.*\d)(?=.*[$@#$!%?&])[A-Za-z\d$@$!%*?&]{8,}$/,
    apellido1: /^[a-zA-Z_\-\ ]{2,}$/,
    apellido2: /^[a-zA-Z_\-\ ]{2,}$/,
    nombre: /^[a-zA-Z_\-\ ]{2,}$/,
    telefono: /^[0-9]{9}$/,
    codigopostal: /^[0-9]{2}-? ?[0-9]{3}$/,
    provincia: /^[a-zA-Z]{2}$/,
    municipios: /^[a-zA-Z]{4}$/,
    direccion: /^[a-zA-Z0-9\.\-_\-\ ]{5,100}$/,
    fechanacimiento: /^[0-9]{4}[-/]{1}[0-9]{2}[-/]{1}[0-9]{2}$/,
    rol: /^[A-Za-z]{4,30}$/,
}
//funcion validar usuarios
function validarUsuario(){
    if (expresiones.usuario.test(usuario.value)) {
        $("#usuario_error").css("visibility", "hidden");
    }else{
        $("#usuario_error").css("visibility", "visible");
        $("#usuario_error").css("color", "red");
    }
}

```

```
function validarNombre(){
    if (expresiones.nombre.test(nombre.value)) {
        $("#nombre_error").css("visibility", "hidden");
    }else{
        $("#nombre_error").css("visibility", "visible");
        $("#nombre_error").css("color", "red");
    }
}

function validarApellido1(){
    if (expresiones.apellido1.test(apellido1.value)) {
        $("#apellido1_error").css("visibility", "hidden");
    }else{
        $("#apellido1_error").css("visibility", "visible");
        $("#apellido1_error").css("color", "red");
    }
}

function validarApellido2(){
    if (expresiones.apellido2.test(apellido2.value)) {
        $("#apellido2_error").css("visibility", "hidden");
    }else{
        $("#apellido2_error").css("visibility", "visible");
        $("#apellido2_error").css("color", "red");
    }
}

//funcion para validar el email
function validarEmail(){
    if (expresiones.email.test(email.value)) {
        $("#email_error").css("visibility", "hidden");
    }else{
        $("#email_error").css("visibility", "visible");
        $("#email_error").css("color", "red");
    }
}

//funcion que valida la contraseña
function validarContrasena(){
    if (expresiones.contrasena.test(contrasena.value)) {
        $("#contrasena_error").css("visibility", "hidden");
    }else{
        $("#contrasena_error").css("visibility", "visible");
        $("#contrasena_error").css("color", "red");
    }
}

function validarTelefono(){
    if (expresiones.telefono.test(telefono.value)) {
        console.log($("#telefono_error"));
        $("#telefono_error").css("visibility", "hidden");
    }else{
        $("#telefono_error").css("visibility", "visible");
        $("#telefono_error").css("color", "red");
    }
}
```

```

function validarCodicopostal(){
    if(expresiones.codicopostal.test(codicopostal.value)){
        document.getElementById('codicopostal_error').style.visibility = "hidden";
    }else{
        document.getElementById('codicopostal_error').style.visibility = "visible";
        document.getElementById('codicopostal_error').style.color = "red";
    }
}

function validarComunidadAutonoma(){
    if(expresiones.municipio.test(municipio.value)){
        document.getElementById('municipio_error').style.visibility = "hidden";
    }else{
        document.getElementById('municipio_error').style.visibility = "visible";
        document.getElementById('municipio_error').style.color = "red";
    }
}

function validarprovincia(){
    if(expresiones.provincia.test(provincia.value)){
        document.getElementById('provincia_error').style.visibility = "hidden";
    }else{
        document.getElementById('provincia_error').style.visibility = "visible";
        document.getElementById('provincia_error').style.color = "red";
    }
}

//funcion para validar el rol
function validarRol(){
    if (expresiones.rol.test(rol.value)) {
        $("#rol_error").css("visibility", "hidden");

    }else{
        $("#rol_error").css("visibility", "visible");
        $("#rol_error").css("color", "red");
    }
}

function validarProvincia(cpostal){
    let cp_provincias = {
        1: "\u00c1lava", 2: "Albacete", 3: "Alicante", 4: "Almer\u00eda", 5: "\u00c1vila",
        6: "Badajoz", 7: "Baleares", 08: "Barcelona", 09: "Burgos", 10: "C\u00f3ceres",
        11: "C\u00f3diz", 12: "Castell\u00f3n", 13: "Ciudad Real", 14: "C\u00f3rdoba", 15: "Coru\u00f1a",
        16: "Cuenca", 17: "Gerona", 18: "Granada", 19: "Guadalajara", 20: "Guip\u00fa\u00f1azcoa",
        21: "Huelva", 22: "Huesca", 23: "Ja\u00e1n", 24: "Le\u00f3n", 25: "L\u00f3rida",
        26: "La Rioja", 27: "Lugo", 28: "Madrid", 29: "M\u00f3laga", 30: "Murcia",
        31: "Navarra", 32: "Orense", 33: "Asturias", 34: "Palencia", 35: "Las Palmas",
        36: "Pontevedra", 37: "Salamanca", 38: "Santa Cruz de Tenerife", 39: "Cantabria", 40: "Segovia",
        41: "Sevilla", 42: "Soria", 43: "Tarragona", 44: "Teruel", 45: "Toledo",
        46: "Valencia", 47: "Valladolid", 48: "Vizcaya", 49: "Zamora", 50: "Zaragoza",
        51: "Ceuta", 52: "Melilla"
    };
    if(cpostal.length == 5 && cpostal <= 52999 && cpostal >= 1000)
        return cp_provincias[parseInt(cpostal.substring(0,2))];
    else
        return "----";
}

```

```

function validarComunidad(cpostal){
    let cp_comunidad = {
        1: "Pa\u00EDs Vasco", 2: "Castilla-La Mancha", 3: "Comunidad Valenciana", 4: "Andaluc\u00EDa", 5: "Castilla y Le\u00F3n",
        6: "Extremadura", 7: "Islas Baleares", 8: "Catalu\u00F1a", 9: "Castilla y Le\u00F3n", 10: "Extremadura",
        11: "Andaluc\u00EDa", 12: "Comunidad Valenciana", 13: "Castilla-La Mancha", 14: "Andaluc\u00EDa", 15: "Galicia",
        16: "Castilla-La Mancha", 17: "Catalu\u00F1a", 18: "Andaluc\u00EDa", 19: "Castilla-La Mancha", 20: "Pa\u00EDs Vasco",
        21: "Andaluc\u00EDa", 22: "Arag\u00E1n", 23: "Andaluc\u00EDa", 24: "Castilla y Le\u00F3n", 25: "Catalu\u00F1a",
        26: "La Rioja", 27: "Galicia", 28: "Comunidad de Madrid", 29: "Andaluc\u00EDa", 30: "Regi\u00F3n de Murcia",
        31: "Comunidad de Navarra", 32: "Galicia", 33: "Principado de Asturias", 34: "Castilla y Le\u00F3n", 35: "Islas Canarias",
        36: "Galicia", 37: "Castilla y Le\u00F3n", 38: "Islas Canarias", 39: "Cantabria", 40: "Castilla y Le\u00F3n",
        41: "Andaluc\u00EDa", 42: "Castilla y Le\u00F3n", 43: "Catalu\u00F1a", 44: "Arag\u00E1n", 45: "Castilla-La Mancha",
        46: "Comunidad Valenciana", 47: "Castilla y Le\u00F3n", 48: "Pa\u00EDs Vasco", 49: "Castilla y Le\u00F3n", 50: "Arag\u00E1n",
        51: "Ceuta", 52: "Melilla"
    };
    if(cpostal.length == 5 && cpostal <= 52999 && cpostal >= 1000)
        return cp_comunidad[parseInt(cpostal.substring(0,2))];
    else
        return "----";
}
codigopostal.onkeyup = function(){
    municipios.value = validarComunidad(codigopostal.value);
    provincia.value = validarProvincia(codigopostal.value);
}
//Funcion que validar dni y la letra del dni
function validardni() {
    let numero;
    let letr;
    let letra;
    let dn = dni.value;
    dni.value = dni.value.slice(0, 8) + dni.value.charAt(8).toUpperCase();
    if(expresiones.dni.test (dn) == true){
        numero = dn.substr(0,dn.length-1);
        letr = dn.substr(dn.length-1,1);
        numero = numero % 23;
        letra='TRWAGMYFPDXBNJZSQVHLCKET';
        letra=letra.substring(numero,numero+1);
        if (letr!=letr.toUpperCase()) {
            document.getElementById("dni_error").style.visibility = "visible";
            document.getElementById("dni_error").style.color = "red";
        }else{
            document.getElementById("dni_error").style.visibility = "hidden";
        }
    }else{
        document.getElementById("dni_error").style.visibility = "visible";
        document.getElementById("dni_error").style.color = "red";
    }
}
function validarDireccion(){
    if(expresiones.direccion.test(direccion.value)){
        document.getElementById('direccion_error').style.visibility = "hidden";
    }else{
        document.getElementById('direccion_error').style.visibility = "visible";
        document.getElementById('direccion_error').style.color = "red";
    }
}
function validarFechaNacimiento(){
    if(expresiones.fechanacimiento.test(fechanacimiento.value)){

```

```
function validarFechaNacimiento(){
    if(expresiones.fechanacimiento.test(fechanacimiento.value)){
        document.getElementById('fechanacimiento_error').style.visibility = "hidden";
    }else{
        document.getElementById('fechanacimiento_error').style.visibility = "visible";
        document.getElementById('fechanacimiento_error').style.color = "red";
    }
}

// funcion para el boton que almacena todas las funciones
function pulsarBoton(){
    validarUsuario();
    validarEmail();
    validarContrasena();
    validarApellido1();
    validarApellido2();
    validarNombre();
    validardni();
    validarTelefono();
    validarCodigopostal();
    validarProvincia();
    validarComunidadAutonoma();
    validarDireccion();
    validarFechaNacimiento();
    validarRol();
}

//oyentes de eventos
usuario.addEventListener("keyup", validarUsuario);
usuario.addEventListener("blur", validarUsuario);
email.addEventListener("keyup", validarEmail);
email.addEventListener("blur", validarEmail);
dni.addEventListener("keyup", validardni);
dni.addEventListener("blur", validardni);
nombre.addEventListener("keyup", validarNombre);
nombre.addEventListener("blur", validarNombre);
apellido1.addEventListener("keyup", validarApellido1);
apellido1.addEventListener("blur", validarApellido1);
apellido2.addEventListener("keyup", validarApellido2);
apellido2.addEventListener("blur", validarApellido2);
codigopostal.addEventListener("keyup", validarCodigopostal);
codigopostal.addEventListener("blur", validarCodigopostal);
provincia.addEventListener("keyup", validarProvincia);
provincia.addEventListener("blur", validarProvincia);
comunidadautonoma.addEventListener("keyup", validarComunidadAutonoma);
comunidadautonoma.addEventListener("blur", validarComunidadAutonoma);
telefono.addEventListener("keyup", validarTelefono);
telefono.addEventListener("blur", validarTelefono);
contrasena.addEventListener("keyup", validarContrasena);
contrasena.addEventListener("blur", validarContrasena);
direccion.addEventListener("keyup", validarDireccion);
direccion.addEventListener("blur", validarDireccion);
fechanacimiento.addEventListener("keyup", validarFechaNacimiento);
fechanacimiento.addEventListener("blur", validarFechaNacimiento);
rol.addEventListener("keyup", validarRol);
rol.addEventListener("blur", validarRol);
```

Registro:

```
//constantes que almacenan los identificadores
const usuario = document.getElementById("usuario");
const email = document.getElementById("email");
const boton = document.getElementById("boton");
//ocultamos los mensajes de errores
document.getElementById("usuario_error").style.visibility = "hidden";
document.getElementById("email_error").style.visibility = "hidden";
const expresiones = {
    usuario : /^[a-zA-Z0-9_\-\ ]{6,}$/,
    email : /^[a-zA-Z0-9_\.+-\S]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-\.]+\$/}
};

//funcion para validar el usuario
function validarUsuario(){
    if (expresiones.usuario.test(usuario.value)) {
        document.getElementById("usuario_error").style.visibility = "hidden";
    }else{
        document.getElementById("usuario_error").style.visibility = "visible";
        document.getElementById("usuario_error").style.color = "red";
    }
}

//constante para validar el email
function validarEmail(){
    if (expresiones.email.test(email.value)) {
        document.getElementById("email_error").style.visibility = "hidden";
    }else{
        document.getElementById("email_error").style.visibility = "visible";
        document.getElementById("email_error").style.color = "red";
    }
}

//funcion para el boton
function pulsarBoton(){
    validarUsuario();
    validarEmail();
}

//oyentes de eventos
usuario.addEventListener("keyup", validarUsuario);
usuario.addEventListener("blur", validarUsuario);
email.addEventListener("keyup", validarEmail);
email.addEventListener("blur", validarEmail);
boton.addEventListener("click", pulsarBoton);
```

Contraseña nueva:

```
//constantes que almacenan los identificadores del html
const password = document.getElementById("password");
const Rcontrasena = document.getElementById("Rcontrasena");
//ocultamos los mensajes de errores
document.getElementById("Rcontrasena_error").style.visibility = "hidden";
document.getElementById("contrasena_error").style.visibility = "hidden";
//constantes para las expresiones regulares
const expresiones = {
    password: /^(?=.{8,}$(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])(?=.*\W)).*/$/
}
//funcion para validar la contraseña
function validarContrasena(){
    if (expresiones.password.test(password.value)) {
        document.getElementById("contrasena_error").style.visibility = "hidden";
    }else{
        document.getElementById("contrasena_error").style.visibility = "visible";
        document.getElementById("contrasena_error").style.color = "red";
    }
}
//funcion para que las contraseñas sean iguales
function validarContrasena2(){
    if (Rcontrasena.value==password.value) {
        document.getElementById("Rcontrasena_error").style.visibility = "hidden";
    }else{
        document.getElementById("Rcontrasena_error").style.visibility = "visible";
        document.getElementById("Rcontrasena_error").style.color = "red";
    }
}
//funcion para el boton que almacena todas las funciones
function pulsarBoton(){
    validarContrasena();
    validarContrasena2();
}
//oyentes de eventos
password.addEventListener("keyup", validarContrasena);
password.addEventListener("blur", validarContrasena);
Rcontrasena.addEventListener("keyup", validarContrasena2);
Rcontrasena.addEventListener("blur", validarContrasena2);
boton.addEventListener("click", pulsarBoton);
```

Jerarquía de directorios

```
debian@debianLAMP:/var/www/html/Codigo$ tree -d
.
├── img
│   └── logo
├── js
│   └── glider
│       ├── docs
│       │   ├── assets
│       │   │   ├── css
│       │   │   │   └── images
│       │   │   ├── fonts
│       │   │   └── js
│       │   └── sass
│           ├── base
│           ├── components
│           ├── layout
│           └── libs
└── php
    └── BD
├── sql
└── styles

20 directories
debian@debianLAMP:/var/www/html/Codigo$ _
```

Contenidos directorios

- ./** -> Contiene los ficheros html y varios archivos .php
- img** -> Contiene todas las fotos de la página.
- js** -> Contiene los archivos javascript para validar los formularios.
- php** -> Ahí es donde se encuentran todos los archivos .php de nuestra página.
- BD** -> Aquí se encuentra el ConectorBD y todos los DAO para que se contente a la BBDD.
- sql**-> Directorio donde se encuentra el .sql para conectarse.
- styles**->Directorio donde se encuentran todos los estilos de la página.

Diseño de la interface

En base al apartado Prototipo web y boceto de la estructura describimos el contenido de la interface de nuestra aplicación.

El sitio web consta de tres partes fundamentales. Esta estructura es común en todas las páginas que forman el sitio web.

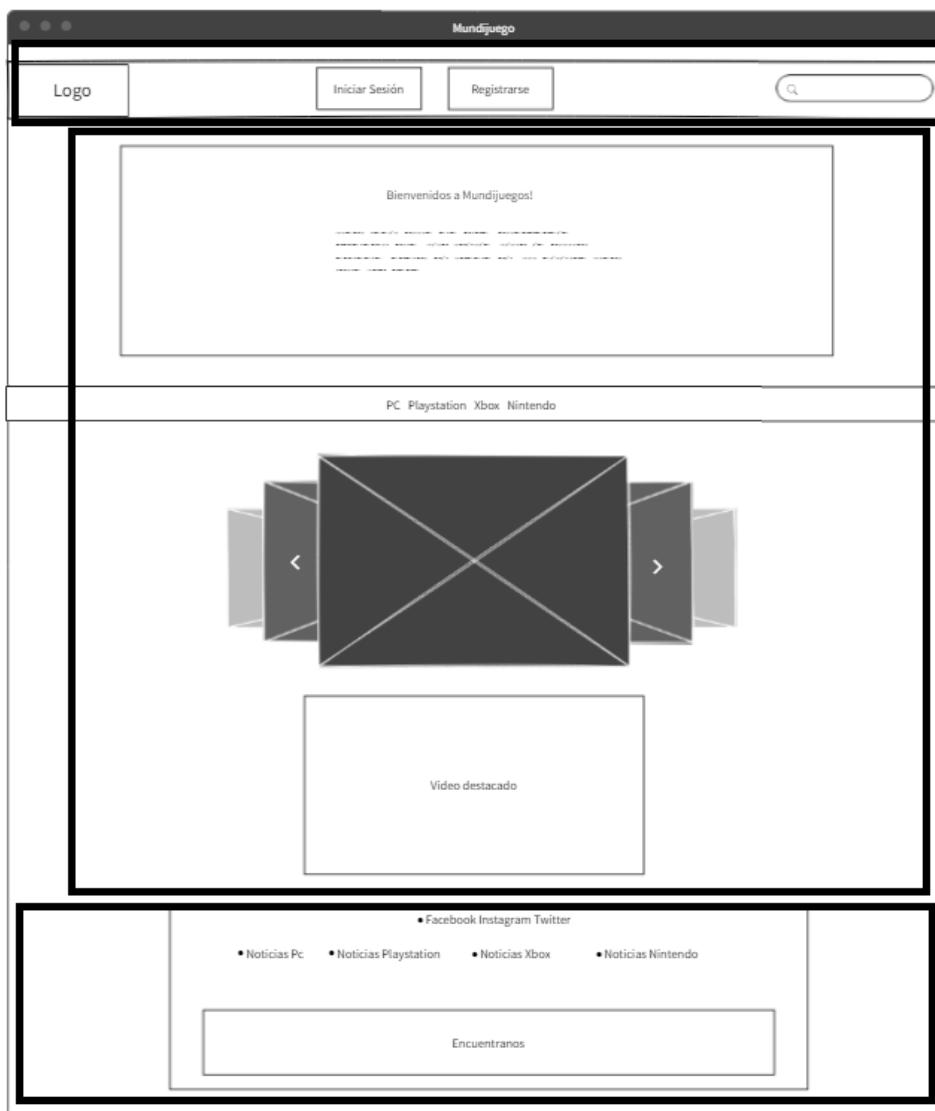
Header: Contiene el logo. También contiene la barra de navegación para ir a las páginas que cambiarán dinámicamente si el usuario inicia sesión.

Body: Se actualizará acorde a la página que visitemos.

Footer: Siempre se encontrar en la página principal.

Estructura gráfica de la interface

Página Web

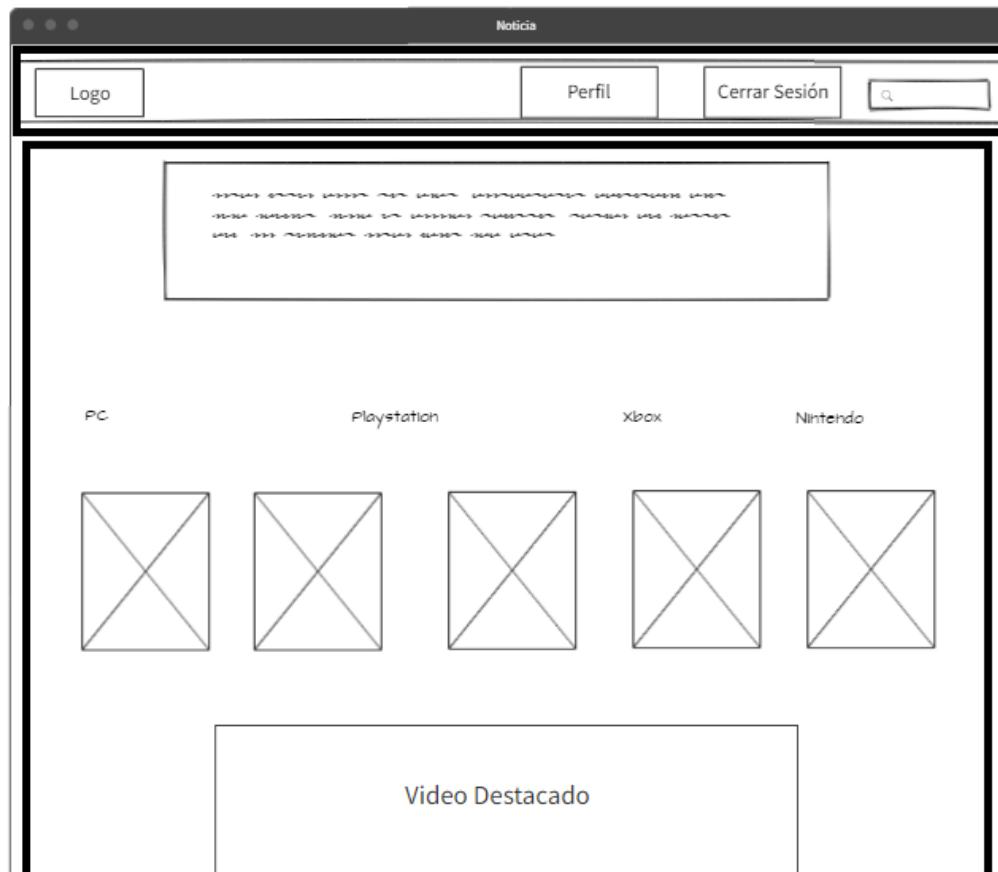


Header

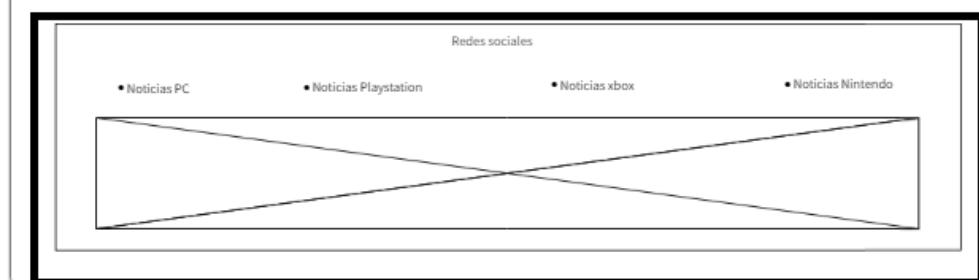
Body

Footer

Zona Admin

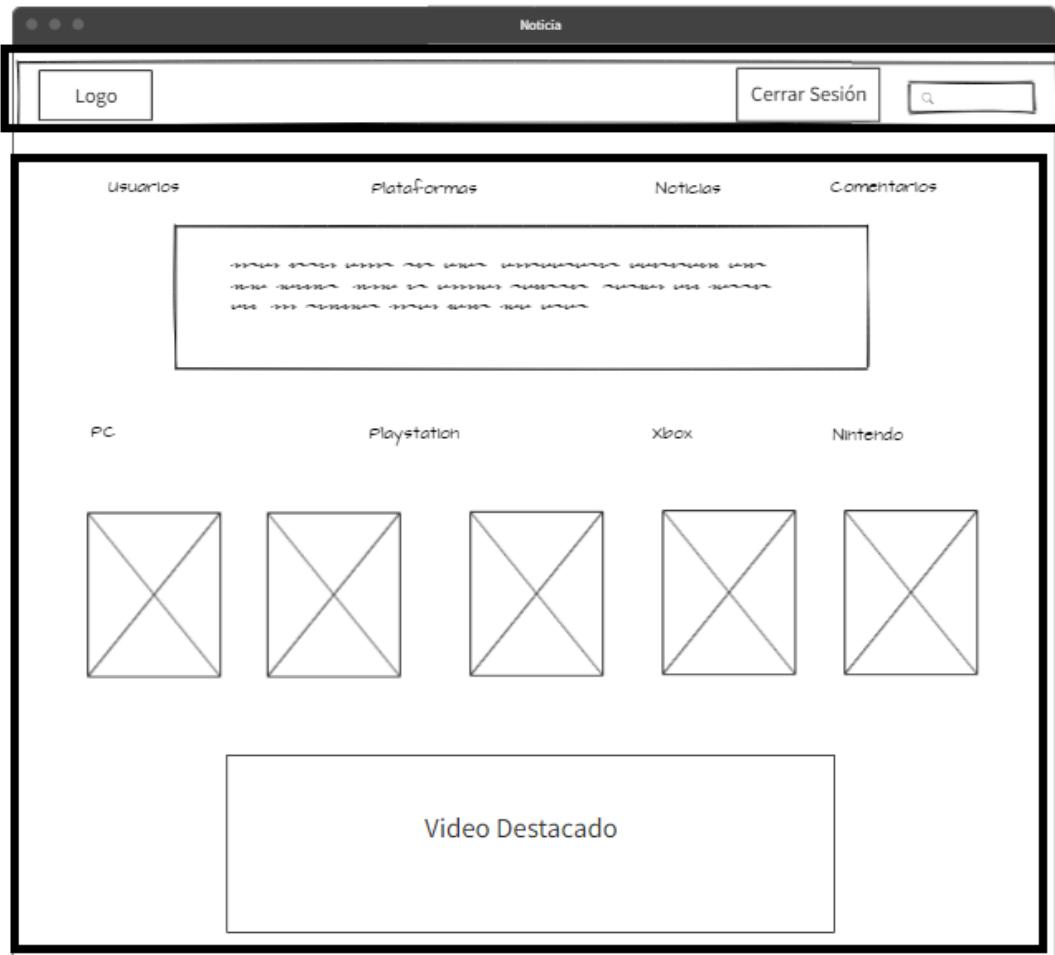


Header



Footer

Zona Usuario:



Header

Body

Footer

IMPLEMENTACIÓN

Casi todos los archivos de la página se implementan con consultas .php

REQUISITO 1 : El usuario se podrá registrar y loguear.

El usuario se puede crear una cuenta e iniciar sesión en ella.

Login:

```
<div class="container1">
<div class="row m-0 justify-content-center align-items-center vh-100">
  <form action="php/comprobar_usuario.php" method="POST">
    <div class="card">
      <div class="card-body">
        <label for="Usuario">Usuario</label><br>
        <input type="text" class="form-control" id="usuario" placeholder="Alumno" name="usuario"><br>
        <span id="usuario_error" class="error">El formato es incorrecto</span><br>

        <label for="">Password</label>
        <input type="password" class="form-control" id="Password" placeholder="Password" name="password"><br>
        <span id="contraseña_error" class="error">La contraseña debe tener mínimo 8 caracteres y al menos una mayúscula, un número y un carácter especial</span><br>
        <a href="contraseña_nueva.html">He olvidado mi contraseña</a>
        <button type="submit" class="btn btn-primary">Enviar</button>
      </div>
    </div>
  </form>
</div>
```

Este es el .html de la página del login, a este .html le sigue un .php

```
<?php
//MOSTRAR ERRORES PHP
ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
error_reporting(E_ALL);
?>
<?php
//Cargamos los archivos que vamos a usar

require "BD/conectorBD.php";
require "BD/DAOUsuarios.php";

$usuario = $_POST["usuario"];
$password = $_POST["password"];
//Nos conectamos a la base de datos y a la consulta

$conexion = conectar(false);
$consulta = consultaLogin($conexion, $usuario, $password);

//Recorremos la consulta

if(mysqli_num_rows($consulta) == 1)
{
  $fila = mysqli_fetch_array($consulta);
  crearSesion($fila);
  header("Location: index.php");
}
else
{
  $consulta2 = consultaUsuarios($conexion, $usuario);
  if(mysqli_num_rows($consulta2) == 1)
  {
    header("Location: ../contraseña_nueva.html");
  }
  else
  {
    header("Location: ../ingresar_usuario.php");
  }
}
```

Este php lo que hace es conectarse a la BBDD para recoger los datos, para que el usuario inicie sesión conectándolo así con DAOUsuarios donde se encuentran todas las consultas necesarias.

Registrarse:

Este es el .html de la página de registro, a este .html le sigue un .php

```
<div class="container">
<form action="php/registro.php" method="POST" validate>
<div class="cardformulario">
    <label for="Usuario:</label><br>
    <input type="text" style="text-align:center" class="form-control" id="usuario" name="usuario" placeholder="Alumno98"><br>
    <span id="usuario_error" class="error">Formato Incorrecto</span><br>

    <label for="Nombre:</label><br>
    <input type="text" style="text-align:center" class="form-control" id="nombre" name="nombre" placeholder="Nombre"><br>
    <span id="nombre_error" class="error">Formato Incorrecto</span><br>

    <label class="form-label">Contraseña: </label>
    <input type="password" style="text-align:center" name="contrasena" class="form-control" placeholder="La contraseña debe tener un minimo de 8 caracteres, contener una mayuscula y un caracter especial."><br>
    <span id="contrasena_error" class="error">Formato Incorrecto</span><br>

    <label for="Primer Apellido:</label>
    <input type="text" style="text-align:center" class="form-control" id="apellido1" name="apellido1" placeholder="1º apellido"><br>
    <span id="apellido1_error" class="error">Formato Incorrecto</span><br>

    <label for="Segundo Apellido:</label>
    <input type="text" style="text-align:center" class="form-control" id="apellido2" name="apellido2" placeholder="2º apellido"><br>
    <span id="apellido2_error" class="error">Formato Incorrecto</span><br>

    <label>Email:</label>
    <input type="email" style="text-align:center" class="form-control" id="email" name="email" placeholder="Email">
    <span id="email_error" class="error">Formato Incorrecto</span><br>

    <label>DNI:</label>
    <input type="text" style="text-align:center" class="form-control" id="dni" name="dni" placeholder="DNI">
    <span id="dni_error" class="error">Formato Incorrecto</span><br>

    <label class="form-label">Direccion: </label>
    <input type="text" style="text-align:center" name="direccion" id="direccion" class="form-control" placeholder="Calle:..."><br>
    <span id="direccion_error" class="error">Formato Incorrecto</span><br>

    <label class="form-label">Fecha de nacimiento: </label>
    <input type="date" style="text-align:center" name="fechanacimiento" id="fechanacimiento" class="form-control" placeholder="Escribe la fecha con numeros: 0000/00/0"><br>
    <span id="fechanacimiento_error" class="error">Formato Incorrecto</span><br>

    <label for="Telefono:</label>
    <input type="text" style="text-align:center" name="telefono" id="telefono" class="form-control" placeholder="Ejemplo: 659777894" pattern="^([0-9]{9})$"><br>
    <span id="telefono_error" class="error">Formato Incorrecto</span><br>

    <label class="form-label">Codigo Postal: </label>
    <input type="text" style="text-align:center" name="codigopostal" id="codigopostal" class="form-control" placeholder="Ejemplo: 52005"><br>
    <span id="codigopostal_error" class="error">Formato Incorrecto</span><br>

    <label class="form-label" for="provincias">Provincia: </label>
    <select name="provincia" style="text-align:center" id="provincia">

    <?php
        require "./php/BD/conectorBD.php";
        require "./php/BD/DAOUsuarios.php";
        $conexion = conectar(false);
        $consulta = obtenerProvincias($conexion);
        echo '<option value="0">Elige una provincia</option>';
        while($fila = mysqli_fetch_assoc($consulta)){
            echo '<option value="'.$fila["id"].'>'.$fila["provincia"].'</option>';
        }
    <?
    </select>
    <span id="provincia_error" class="error">Formato Incorrecto</span><br>

    <label class="form-label">Municipio: </label>
    <div id="listaMunicipios" style="text-align:center">
        <span id="municipio_error" class="error">Formato Incorrecto</span><br>

    </div>
    <br>

    <button type="submit" class="btn btn-primary">Enviar</button>
    </div>
    </div>
    </div>
    <br><br>
</div>
<!-- AQUI EL JS QUE HAGA QUE TARDE EN CARGAR LA WEB -->
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DFXd2htPH0lsS5s5CTpuj/zY4C+0GpamoFVy38MV8nE+IbbVVUew+OrCxRkfj" crossorigin="anonymous"></script>
<script src="https://code.jquery.com/jquery-3.6.0.js" integrity="sha256-H+K7U5CnX1h5ywQfKtSJ8PCmoN9eaq30g0h27Xc0jk=" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js" integrity="sha384-9/reFTGAW83EW2RDzSU50Kaizap3H61zH81Poy1Fhb6U+6Zp6G7niu735Sk7IN" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.min.js" integrity="sha384-wlQ4orYjBQdnkGMiLwBQzv7s" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ho+j7jyWk8FNQe+A12Hb8Rrq26LrZ/JpcUGGOn+Y7RsweNrtN/tE3MoK7ZeZDyx" crossorigin="anonymous"></script>
<script src="http://1000hz.github.io/bootstrap-validator/dist/validator.min.js"></script>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ho+j7jyWk8FNQe+A12Hb8Rrq26LrZ/JpcUGGOn+Y7RsweNrtN/tE3MoK7ZeZDyx" crossorigin="anonymous"></script>
<script src="http://1000hz.github.io/bootstrap-validator/dist/validator.min.js"></script>
<script src="js/registro.js"></script>
<script>
    $(document).ready(function(){
        recargarLista();
        $('#provincia').change(function(){
            recargarLista();
        });
    });

    </script>
    <script>
        function recargarLista(){
            $.ajax({
                type:"POST",
                url:"obtenerMunicipios.php",
                data: "input="+$( "#provincia").val() ,

                success:function(data){
                    $('#listaMunicipios').html(data);
                }
            })
        }
    </script>
</script>
```

Este php lo que hace es conectarse a la BBDD para recoger los datos, para que el usuario inicie sesión conectándolo así con DAOUsuarios donde se encuentran todas las consultas necesarias.

```
<?php
ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
error_reporting(E_ALL);
?>
<?php

//Cargamos los archivos que vamos a usar
require 'BD/conectorBD.php';
require 'BD/DAOUsuarios.php';

//Usamos las variables que vamos a coger
$Usuario = $_POST['usuario'];
$Nombre = $_POST['nombre'];
$Password = $_POST['contraseña'];
$Apellido1 = $_POST['apellido1'];
$Apellido2 = $_POST['apellido2'];
$Email = $_POST['email'];
$DNI = $_POST['dni'];
$Telefono = $_POST['telefono'];
$CP = $_POST['codigopostal'];
$Provincia = $_POST['provincia'];
$Municipio = $_POST['municipio'];
$Direccion = $_POST['direccion'];
$FechaNacimiento = $_POST['fechanacimiento'];

//Nos conectamos a la base de datos y a la consulta
$conexion = conectar(false);
$consulta = insertarUsuarios($conexion,$Usuario,$Password,$Email,$DNI,$Apellido1,$Apellido2,$Municipio,$Nombre,$Telefono,$CP, $Direccion, $Provincia, $FechaNacimiento);

//Recorremos la consulta
if($consulta){

    header('Location: ../login.html');

} else {
    var_dump($_POST);
}

?>
```

REQUISITO 2 : Si el usuario no cierra la sesión se queda logueada, es decir hasta que no cierra la sesión no sale.

Siempre se queda la sesión iniciada a no ser que el usuario decida cerrarla, para ello utilizamos este código:

```
1  <?php
2      session_start();
3      session_destroy();
4      header('Location: ../login.html');
5  ?>
```

REQUISITO 3 : Solo los usuarios registrados pueden ver la noticia entera y comentar en ella.

Si el usuario tiene un ROL mostrará la noticia y la opción de comentar en ella.

Para que muestre solo lo que tiene un ROL lo definimos así:

```
<?php  
| if($_SESSION['Rol']){  
?>
```

REQUISITO 4 : Los usuarios no registrados solo podrán ver el título de la noticia y mitad de la descripción.

Si no tienen rol el usuario verá la página normal.

REQUISITO 5: El usuario registrado tendrá la opción de modificar su perfil, también tendrá la opción de recuperar la contraseña, si no se acuerda de ella y eliminar su perfil.

Este es el .html de la página de modificar, a este .html le sigue un .php

```
<div class="row m-0 justify-content-center align-items-center">  
<div class="card">  
  <form action="actualizarperfil.php" method="POST" enctype="multipart/form-data">  
    <label>Usuario</label><br>  
    <input type="text" class="" name="Usuario" require><br>  
    <label>Password</label><br>  
    <input type="password" class="" name="Password" require><br>  
    <label>Nombre</label><br>  
    <input type="text" class="" name="Nombre" require><br>  
    <label>Primer Apellido</label><br>  
    <input type="text" class="" name="Apellido1" require><br>  
    <label>Segundo Apellido</label><br>  
    <input type="text" class="" name="Apellido2" require><br>  
    <label>Telefono</label><br>  
    <input type="number" class="" name="Telefono" require><br>  
    <label>Email</label><br>  
    <input type="email" class="" name="DNI" require><br>  
    <label>DNI</label><br>  
    <input type="text" class="" name="Email" require><br>  
    <label>Direccion</label><br>  
    <input type="text" class="" name="Direccion" require><br>  
    <label>CP</label><br>  
    <input type="number" class="" name="CP" require><br>  
    <label>Provincia</label><br>  
    <input type="text" class="" name="Provincia" require><br>  
    <label>Comunidad Autonoma</label><br>  
    <input type="text" class="" name="ComunidadAutonoma" require><br>  
    <label>Fecha de Nacimiento</label><br>  
    <input type="date" id="start" name="FechaNacimiento" value="2018-07-22" min="2018-01-01" max="2018-12-31"><br><br>  
  
    <button type="submit" class="btn btn-primary">Modificar</button><br>  
    <input type="hidden" name="idUsuario" value="<?php echo $idUsuario?>">  
  </form>  
</div>  
</div>
```

Este php lo que hace es conectarse a la BBDD para recoger los datos, para que el usuario inicie sesión conectándolo así con DAOUsuarios donde se encuentran todas las consultas necesarias.

REQUISITO 6: El usuario registrado podrá ver las noticias más comentadas.

```
<?php
if($_SESSION['Rol']){
?>
<section class="titulo">
<div>
| <h2><span>! Noticias con más comentarios! </span></h2>
</div>
</section>
<div class="glider-contain">
<div class="glider">
<?php
$conexion = conectar(false);
$consulta = mostrarNoticiasPorComentarios($conexion);
while ($noticiasPorComentarios = mysqli_fetch_assoc($consulta)){
?>
<div class="card" style="width: 18rem;">
<?php
$consultaTituloNoticia = mostrarTituloNoticia($conexion,$noticiasPorComentarios['idNoticias']);
while ($tituloNoticia = mysqli_fetch_assoc($consultaTituloNoticia)){
?>
<img src=<?= $tituloNoticia['Imagen'];?> class="card-img-top" alt="...">
<div class="card-body">
<h5 class="card-title"><?= $tituloNoticia['Titulo'];?></h5>
<?php ?>
<a href="detallesnoticia.php?id=<?php echo $noticiasPorComentarios['idNoticias']; ?>" class="btn btn-primary">Leer mas <i class="fas fa-angle-double-right"></i></a>
</div>
<?php ?>
</div>
<button aria-label="Previous" class="glider-prev"></button>
<button aria-label="Next" class="glider-next"></button>
<div role="tablist" class="dots"></div>
```

Solo el usuario registrado podrá ver las noticias mas comentadas en la página, dándole así al usuario un privilegio.

REQUISITO 7: Los administradores tienen un panel de administración.

```
<?php
//Cargamos los archivos que vamos a usar
require 'BD/DAOUsuarios.php';
require 'BD/conectorBD.php';
//Nos conectamos a la base de datos
$conexion = conectar(false);

//Usamos las variables que vamos a coger
$usuario= $_POST['usuario'];
$password = $_POST['password'];
$nombre = $_POST['nombre'];
$Apellido1= $_POST['apellido1'];
$Apellido2 = $_POST['apellido2'];
$Telefono = $_POST['telefono'];
$Email = $_POST['email'];
$CodigoPostal = $_POST['codigopostal'];
$Provincia = $_POST['provincia'];
$ComunidadAutonoma = $_POST['municipio'];
$FechaNacimiento = $_POST['fechanacimiento'];
$Direccion = $_POST['direccion'];
$dni = $_POST['dni'];
$idUsuario = $_POST['idUsuario'];

//Nos conectamos a la consulta con la variable y los datos que necesitamos
$consulta= modificarPerfil($conexion, $usuario, $password, $nombre, $Apellido1, $Apellido2, $Telefono, $Email, $CodigoPostal, $Provincia, $ComunidadAutonoma, $FechaNacimiento, $Direccion, $dni, $idUsuario);
if($consulta){
    $consulta2=consultarUsuarios($conexion, $usuario);
    $usuario=mysqli_fetch_assoc($consulta2);
    crearSesion($usuario);
    header('Location: perfil.php');

} else {
    header('Location: modificarPerfil.php');
}
?>
```

```

<?php
if(isset($_SESSION["Rol"])) {
    if($_SESSION["Rol"]=="admin") {?>

        <div class="adminpanel">
            <section class="titulo">
                <hr>
                <div><h2><span>↓ Panel de administración </span></h2></div>
            </section>
            <a href="panel.php" class="actionadmin">Usuarios</a>
            <a href="adminplataforma.php" class="actionadmin">Plataformas</a>
            <a href="adminnoticias.php" class="actionadmin">Noticias</a>
            <a href="comprobar_comentario.php" class="actionadmin">Comentarios</a>
        </div>
        <hr>

    <?php } ?>

```

De esta forma los administradores podrán moverse mejor por la página teniendo así un panel de acceso rápido.

REQUISITO 8: Los administradores pueden ver los datos de los usuarios, cambiar su rol y poder eliminarlos.

```

<?php
//Nos conectamos a la base de datos y mostramos en este caso los usuarios

$conexion = conectar(false);
$result = mostrarUsuario($conexion);

while($fila=mysqli_fetch_assoc($result)){?>

<tr>
    <?php
        foreach($fila as $key => $value){?>
            <td> <?= $value ?> </td>
            <?php
            |   ?
            |   ?>
    </tr>
    <?php
    |
    ?>

<td><button ><a href="modificarusuario.php?idUsuario= <?php echo $fila['idUsuario'];?>" value="modificar" name="modificar">Cambiar Rol</button></td>
<td><button ><a href="eliminarUsuario.php?idUsuario= <?php echo $fila['idUsuario'];?>" value="eliminar" name="eliminar">Eliminar</button></td>

</tr>
    <?php
    |
    ?>

```

Con esta consulta mostramos los usuarios logueados en la página con todos sus datos. También el administrador puede cambiar su rol:

```

<?php
//Cargamos los archivos que vamos a usar

require "BD/DAOUsuarios.php";
require "BD/conectorBD.php";
//Nos conectamos a la base de datos

$conexion = conectar(false);
//Usamos las variables que vamos a coger

$rol = $_POST['Rol'];
$idUsuario = $_POST['idUsuario'];
//Nos conectamos a la consulta con la variable y los datos que necesitamos

$consulta = modificarusuario($conexion, $rol, $idUsuario);

header('Location: panel.php');

?>

```

REQUISITO 9: Solo el administrador podrá hacer a un usuario normal admin.

Solo el administrador podrá cambiar el rol del usuario.

```
<?php
//Cargamos los archivos que vamos a usar

require "BD/DAOusuarios.php";
require "BD/conectorBD.php";
//Nos conectamos a la base de datos

$conexion = conectar(false);
//Usamos las variables que vamos a coger

$rol = $_POST['Rol'];
$idUsuario = $_POST['idUsuario'];
//Nos conectamos a la consulta con la variable y los datos que necesitamos

$consulta = modificarusuario($conexion, $rol, $idUsuario);

header('Location: panel.php');

?>
```

REQUISITO 10: Los administradores pueden ver las plataformas que hay en la página, insertar nuevas plataformas, modificarlas y eliminarlas.

Mostramos las plataformas, haciendo la consulta necesaria.

```
<?php
//Cargamos los archivos que vamos a usar

require "BD/conectorBD.php";
require "BD/DAOplataforma.php";
//Nos conectamos a la base de datos y a la consulta

$conexion = conectar(false);
$consulta = mostrarconsola($conexion);
//Recorremos la consulta

while($mostrar=mysqli_fetch_array($consulta)){
?>

    <tr>
        <td><?php echo $mostrar['idPlataforma'] ?></td>
        <td><?php echo $mostrar['Nombre'] ?></td>
        <td><b> <img src=<?php echo $mostrar['Imagen']?>> width="120"> </b></td>
        <td> <a href="modificar.php?IdPlataforma= <?php echo $mostrar['idPlataforma'];?>>" value="modificar" name="modificar">Modificar</button></td>
        <td> <a href="eliminar.php?Idconsola= <?php echo $mostrar['idPlataforma'];?>>" value="eliminar" name="eliminar">Eliminar</button></td>
    </tr>
    <?php
}
?>
```

Y para insertarlas usamos un .html llamando después el .php donde se tiene que insertar.

```
<body>
    <div class="container1">
        <nav class="nav-main1">
            <a href="index.php"></a>
            <a href="php/adminplataforma.php" class="actionpanel">Atrás</a>
        </nav>
    </div>
    <br>
    <div class="row m-0 justify-content-center align-items-center">
        <div class="card">
            <form action=".php/insertar.php" method="POST" enctype="multipart/form-data">
                <a>Nombre</a><br>
                <input type="text" class="" name="Nombre"><br>
                <a>Imagen</a><br>
                <input type="file" class="" name="Imagen"><br>
                <br>
                <button type="submit" class="btn btn-primary">Enviar</button>
            </form>
        </div>
    </div>
```

Para modificarlas utilizamos un botón que nos lleva a un html donde hace su función un .php.

```
<?php
//Cargamos los archivos que vamos a usar

require "BD/DAOplataforma.php";
require "BD/conectorBD.php";
//Usamos las variables que vamos a coger
$Nombre = $_POST['Nombre'];
echo $Nombre;
$ImagenesN=$_FILES['Imagen']['name'];
$imagenA=$_FILES['Imagen']['tmp_name'];
$rutaImg="../../img/".$ImagenesN;
move_uploaded_file($imagenA,$rutaImg);
//Nos conectamos a la base de datos y a la consulta
$conexion = conectar(false);
$consulta = insertarProductos($conexion, $Nombre, $rutaImg);
//Mostramos la localización donde queremos que vayan dirigidos los datos.
if($consulta){

    header('Location: adminplataforma.php');

} else {

    header('Location: ../../insertar.html');

}

?>
```

REQUISITO 11: Los administradores pueden ver las noticias que hay en la página, insertar nuevas noticias, modificarlas y eliminarlas.

Mostramos las noticias, haciendo la consulta necesaria.

```
<?php
//Cargamos los archivos que vamos a usar

require "BD/conectorBD.php";
require "BD/DAONoticia.php";
//Nos conectamos a la base de datos y a la consulta

$conexion = conectar(false);
$consulta = mostrarNoticias($conexion);
//Recorremos la consulta

while($mostrar=mysqli_fetch_array($consulta)){
    ?>

    <tr>
        <td><?php echo $mostrar['Titulo'] ?></td>
        <td><?php echo $mostrar['Descripcion'] ?></td>
        <td><img src=<?php echo $mostrar['Imagen'] ?> width="120"></td>
        <td><?php echo $mostrar['NombreP'] ?></td>
        <td> <button >a href="modificarNoticia.php?IdNoticias= <?php echo $mostrar['idNoticias'];?>" value="modificar" name="modificar">Modificar</button></td>
        <td> <button >a href="eliminarNoticia.php?IdNoticias= <?php echo $mostrar['idNoticias'];?>" value="eliminar" name="eliminar">Eliminar</button></td>
    </tr>
    <?php
}

    ?>
</table>
```

Y para insertarlas usamos un .html llamando después el .php donde se tiene que insertar.

```
</head>
<body>
    <nav class="nav-main1">
        <a href="index.php"></a>
        <a href="php/adminnoticias.php" class="actionpanel">Atrás</a>
    </nav>
    <br>
    <div class="row m-0 justify-content-center align-items-center">
        <div class="card">
            <form action="php/insertarNoticia.php" method="POST" enctype="multipart/form-data">

                <a>Titulo</a><br>
                <input type="text" class="" name="Titulo"><br>
                <a>Número Plataforma</a><br>
                <input type="text" class="" name="Plataforma"><br>
                <a>Nombre Plataforma</a><br>
                <input type="text" class="" name="NombreP"><br>
                <a>Descripción</a><br>
                <input type="text" class="" name="Descripción"><br>
                <a>Imagen</a><br>
                <input type="file" class="" name="Imagen"><br><br>

                <button type="submit" class="btn btn-primary">Insertar</button><br>
            </form>
        </div>
    </div>
</body>
</html>
```

Para modificarlas utilizamos un botón que nos lleva a un html donde hace su función un .php.

```
<?php
//Cargamos los archivos que vamos a usar
require "BD/DAOnoticia.php";
require "BD/conectorBD.php";
//Nos conectamos a la base de datos
$conexion = conectar(false);
//Obtenemos el apartado que queremos utilizar.
$idNoticias = $_GET['IdNoticias'];
$consulta=mostrarNoticiaId($conexion, $idNoticias);
$mostrar=mysqli_fetch_assoc($consulta);

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Modificar Noticia</title>
    <link rel="stylesheet" type="text/css" href="../styles/style2.css">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css" integrity="sha384-TX8t27EcRE3e/iM7zmQxVncDAy5uIKz4rEkgIXeMed4M9j1fIDPvg6uqK12xXr2" crossorigin="anonymous">
    <link rel="icon" href="../img/logo.png">
</head>
<body>
<header>

<div class="container1">
    <nav class="nav-main1">
        <a href="index.php"></a>
        <a href="adminnoticias.php" class="actionpanel">Atrás</a>
    </nav>
</div>
<br>
<div class="row m-0 justify-content-center align-items-center">
<div class="card">
    <form action="actualizarNoticia.php" method="POST" enctype="multipart/form-data">
        <input type="text" class="" id="Titulo" name="Titulo" require><br>
        <input type="text" class="" id="Descripcion" name="Descripcion" require><br>
        <input type="file" class="" id="Imagen" name="Imagen"><br>
        <input type="hidden" name="Imagen_tmp" value=<?php echo $mostrar['Imagen'] ?>">
        <input type="text" class="" id="Plataforma" name="Plataforma" require><br>
        <input type="text" class="" id="NombreP" name="NombreP" require><br>
        <input type="submit" value="Modificar" require>
        <input type="hidden" name="IdNoticias" value=<?php echo $idNoticias ?>">
    </form>
</div>
</div>
</body>

```

REQUISITO 12: Los administradores pueden ver los comentarios, aceptarlos o eliminarlos.

Mostramos los comentarios, haciendo la consulta necesaria. Aquí el administrador podrá aceptar o eliminar el comentario.

```
<table border="1" class="table table-dark">
    <tr>
        <th>idComentarios</th>
        <th>idNoticias</th>
        <th>idUser</th>
        <th>Contenido</th>
        <th>Estado</th>
    </tr>

    <?php
    ini_set('display_errors', 1);
    ini_set('display_startup_errors', 1);
    error_reporting(E_ALL);

    //Cargamos los archivos que vamos a usar

    require "BD/conectorBD.php";
    require "BD/DAOcomentarios.php";

    //Nos conectamos a la base de datos y a la consulta

    $conexion = conectar(false);
    $consulta = mostrarComentariosAprobados($conexion);

    //Recorremos la consulta

    while($mostrar=mysqli_fetch_array($consulta)) {
        ?>

        <tr>
            <td><?php echo $mostrar['idcomentarios'] ?></td>
            <td><?php echo $mostrar['idNoticias'] ?></td>
            <td><?php echo $mostrar['idUser'] ?></td>
            <td><?php echo $mostrar['contenido'] ?></td>
            <td><?php echo $mostrar['estado'] ?></td>
            <td> <button><a href="aceptarcomentario.php?idcomentarios= <?php echo $mostrar['idcomentarios'];?>" value="aceptar" name="aceptar">Aceptar</button></a></td>
            <td> <button><a href="eliminarcomentario.php?idcomentarios= <?php echo $mostrar['idcomentarios'];?>" value="eliminar" name="eliminar">Eliminar</button></a></td>
        </tr>
        <?php
    }
    ?>
</table>
```

REQUISITO 13: La página tendrá diferentes secciones de noticias (PC, Nintendo, Playstation, Xbox).

```
<br>
<nav class="nav-main2">
    <li class="categoria">
        <a href="noticiasvj.php">PC</a>
    </li>
    <li class="categoria">
        <a href="noticiasplay.php">Playstation</a>
    </li>
    <li class="categoria">
        <a href="noticiasxbox.php">Xbox</a>
    </li>
    <li class="categoria">
        <a href="noticiasnintendo.php">Nintendo</a>
    </li>
</ul>
</nav>
<br>
```

REQUISITO 14: La página principal tendrá un carrusel en el que se muestran las consolas, estas se actualizan conforme se vaya actualizando la página.

```
<div class="container">
    <section class="contacto row justify-content-center">
        <div class="col-12 col-md-9 text-center"></div>
        <div id="myCarousel" class="carousel slide" data-ride="carousel">
            <div class="carousel-inner">
                <?php
                    require "BD/DAOplataforma.php";
                    //Creamos la conexión a la BD.
                    $conexion = conectar(false);

                    //Lanzamos la consulta.
                    $consulta = consultaPlataforma($conexion);
                    $i = 0;
                    while($fila = mysqli_fetch_assoc($consulta))
                    {
                        ?>
                        <div class="item <?php echo ($i == 0) ? 'active' : ''?>">
                            
                        </div>
                        <?php
                            $i++;
                        ?>
                    }
                </div>
                <a class="left carousel-control" href="#myCarousel" data-slide="prev"><!--boton para la izquierda -->
                    <span class="glyphicon glyphicon-chevron-left"></span>
                    <span class="sr-only">Anterior</span>
                </a>
                <a class="right carousel-control" href="#myCarousel" data-slide="next"><!--Boton para la derecha-->
                    <span class="glyphicon glyphicon-chevron-right"></span>
                    <span class="sr-only">Siguiente</span>
                </a>
            </div>
        </div>
    <?php
    }>
?>
```

REQUISITO 15: Para acceder a las noticias tendrán que clicar en una de las tarjetas y le mostrará toda la información aportada.

Depende del id de la plataforma saldrán las noticias indicadas a cada plataforma, con el botón de ver mas el usuario podrá acceder a la noticia.

```
<?php

//Cogemos las bases de datos que vamos a utilizar
require 'BD/conectorBD.php';
require 'BD/DAONoticia.php';

//Nos conectamos a la base de datos
$conexion = conectar(false);

//Usamos una función que nos permite mostrar las funciones de la base de datos que vamos a utilizar
$consulta = mostrarNoticias($conexion);

//Recorre la consulta y muestra la información

while ($mostrar=mysqli_fetch_array($consulta)) {

    ?>
    <?php
if($mostrar['Plataforma']=="5"){
?>




##### <b><?php echo $mostrar['Titulo'] ?> </b></h5> <hr> <b><?php echo $mostrar['NombreP'] ?> </b></p>


</div>
<td>
    <ul class="nav justify-content-center">
        <li class="nav-item" id="botones">
            <button type="submit"> <a href="detallesnoticia.php?id=<?php echo $mostrar['idNoticias']; ?>" value="Detalles" name="Detalles" >Ver detalles</a></button></li>
        </ul>
        <br>
    </td>
</div>
<?php
}
?>


```

REQUISITO 16: Diseño responsive en todas las páginas.

Cada vez hay mas usuarios de Internet que utilizan dispositivos móviles para navegar. Si nos fijamos en la analítica de nuestra web, nos damos cuenta que cada vez hay un gran numero de visitas que provienen de dispositivos móviles, por lo que las resoluciones a las cuales vemos nuestro desarrollo son diferentes.

La principal necesidad que existe para hacer un diseño «adaptable», es porque se pierden bastantes visitantes, al no tener el site adaptable a todos.

Para ello se ha empleado Mediaqueries

```
/* Media QUERY 550px*/
@media screen and (max-width: 700px) {
    .container1 {
        padding-right: 0 !important;
        padding-left: 0 !important;
        margin-right: 0 !important;
        margin-left: 0 !important;
    }

    .nav-main1{
        max-width: 100%;
        margin: none;
        background-color: #6684ab;
        display: grid;
        grid-template-columns: 1fr 1fr 1fr;
        grid-template-rows: 1fr 1fr;
        grid-template-areas:
            "elemento1 elemento2 elemento2"
            "elemento3 elemento4 elemento4";
        justify-content: space-between;
        align-items: center;
        justify-items: center;
        grid-auto-flow: column;
    }

    .banner1{
        width: 100%;
        font-size: 12px !important;
    }

    .banner1 h1{
        font-size: 22px !important;
    }

    .actionadmin{
        margin: 15px 10px;
        font-size: 18px !important;
    }

    .nav-main2{
        width: 80%;
        margin: 0 auto;
        display: flex;
        flex-wrap: wrap;
        justify-content: space-around;
    }

    div#myCarousel {
        width: 80%;
        margin: 0 auto;
    }

    iframe{
        width: 400px;
    }

    .row {
        margin-right: 0px !important;
        margin-left: 0px !important;
        justify-content: center;
    }

    .content h1 {
        font-size: 22px !important;
    }
}
```

REQUISITO 17: Control de errores en formularios.

Necesitamos evitar que se pueda introducir cualquier tipo de dato, así como datos erróneos en los formularios (por ejemplo, escribir números en el nombre) y mediante iconos y mensajes de error para ayudar al usuario. Para incluirlo se ha empleado JavaScript.

Para su validación hemos utilizado las expresiones adecuadas para cada campo:

```
const expresiones = {
    usuario: /^[a-zA-Z0-9_\-\-]{6,}$/,
    email: /^[a-zA-Z0-9\.\+\-]+@[a-zA-Z0-9\.\-]+\.[a-zA-Z0-9\.\-]+$/,
    dni: /^[0-9]{8}[A-Za-z]$/,
    contrasena: /^(?=.*[A-Z])(?=.*\d)(?=.*[$@#$!%&])[A-Za-z\d$@$!%*?&]{8,}$/,
    apellido1: /^[a-zA-Z_\-\-]{2,}$/,
    apellido2: /^[a-zA-Z_\-\-]{2,}$/,
    nombre: /^[a-zA-Z_\-\-]{2,}$/,
    telefono: /^[0-9]{9}$/,
    codigopostal: /^[0-9]{2}-?[0-9]{3}$/,
    provincia: /^[a-zA-Z]$/,
    municipios: /^[a-zA-Z]{4,}$/,
    direccion: /^[a-zA-Z0-9\.\s\-\-]{5,100}$/,
    fechanacimiento: /^[0-9]{4}[-]{1}[0-9]{2}[-]{1}[0-9]{2}$/,
    rol: /[A-Za-z]{4,30}/
}
```

PRUEBAS

Anteriormente he estado mostrando todos los requisitos con capturas.

DESPLIEGUE

Variables de entorno:

```
<?php

function conectar($esRemota){

    if($esRemota){
        $servidor = "";

    }else{
        $servidor = "localhost:3307";
        $usuario = "root";
        $password = "root";
        $bd = "final";
    }

    $conector = mysqli_connect($servidor, $usuario, $password, $bd);
    if ($conector ) {
        return $conector;
    }
    else{
        echo mysqli_connect_error();
    }

    echo "conectamos a la BD<br>";

    $conectar = mysqli_connect($servidor, $usuario, $password, $bd);

    if($conectar){
        echo "La conexion se ha realizado con exito<br>";
        return $conectar;
    }else{
        echo "Error no se ha conectado<br>";
        echo mysqli_connect_error();
        exit;
    }
}
```

Nombre del fichero de los datos en el proyecto:

Datos: final.sql

BBDD: final

Se toman los parámetros como USER y PASS de la BBDD

```
# Se toman los parámetros como USER y PASS de la BBDD
if [ $# = 2 ];
then
    USERDB=$1
    PASSDB=$2
fi
```

Copiamos el contenido de la carpeta proyecto a la página html

```
# Copiamos el contenido de la carpeta proyecto a la página html  
cp -r ../Codigo/ $WWW
```

Restauramos los datos de ejemplo a la BBDD

```
# Restauramos los datos de ejemplo a la BBDD  
mysqladmin -u $USERDB -p$USERDB create $BBDD  
mysql -u $USERDB -p$USERDB $BBDD < ../DataBase/$DATOS
```

Mostramos url de carga

```
# Mostramos url de carga  
echo "http://$HOST/Codigo/php/index.php"
```

Cambiamos los permisos de las carpetas que contienen las imágenes

```
sudo chmod +777 /var/www/html/Codigo/img
```

HERRAMIENTAS

Descripción de todas las herramientas que se han usado para el desarrollo del proyecto.

Para la realización del proyecto se han empleado las siguientes herramientas:

Visual Studio Code: Es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.

Se ha utilizado para crear el código y editararlo.

MAMP: El acrónimo MAMP se refiere al conjunto de programas software comúnmente usados para desarrollar sitios web dinámicos sobre sistemas operativos Apple Macintosh, MAC OS X. - macOS: Sistema operativo. - Apache: Servidor Web. - MySQL: Sistema de gestión de bases de datos.

Lo he utilizado como servidor Apache y como soporte para MySQL Workbench para la BD.

MySQL Workbench: MySQL Workbench es una herramienta visual de diseño de bases de datos que integra desarrollo de software, administración de bases de datos, diseño de bases de datos, gestión y mantenimiento para el sistema de base de datos MySQL.

He utilizado MySQL para la creación de la BD, las tablas y para poder crear sus relaciones entre ellas.

OVA Lamp:

OVA es un Objeto Virtual de Aprendizaje, que algunas empresas ponen en funcionamiento y suben a internet para ser usadas.

LAMP: es el acrónimo usado para describir un sistema de infraestructura de internet que usa las siguientes herramientas: Linux, Apache, MySQL/MariaDB y PHP.

Se utiliza como máquina de pruebas de nuestro proyecto para instalar el SCRIPT de despliegue y probar el despliegue de nuestra aplicación.

LENGUAJES

Descripción de los lenguajes y frameworks utilizados para el desarrollo del proyecto.

Para la realización del proyecto se han empleado los siguientes lenguajes.

HTML: Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros.

Utilizado para crear las páginas de nuestra web.

PHP: Es un lenguaje de programación de uso general que se adapta especialmente al desarrollo web.

Utilizado para crear las páginas de nuestra web.

JavaScript: Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza para validar los formularios.

Bootstrap: Es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web.

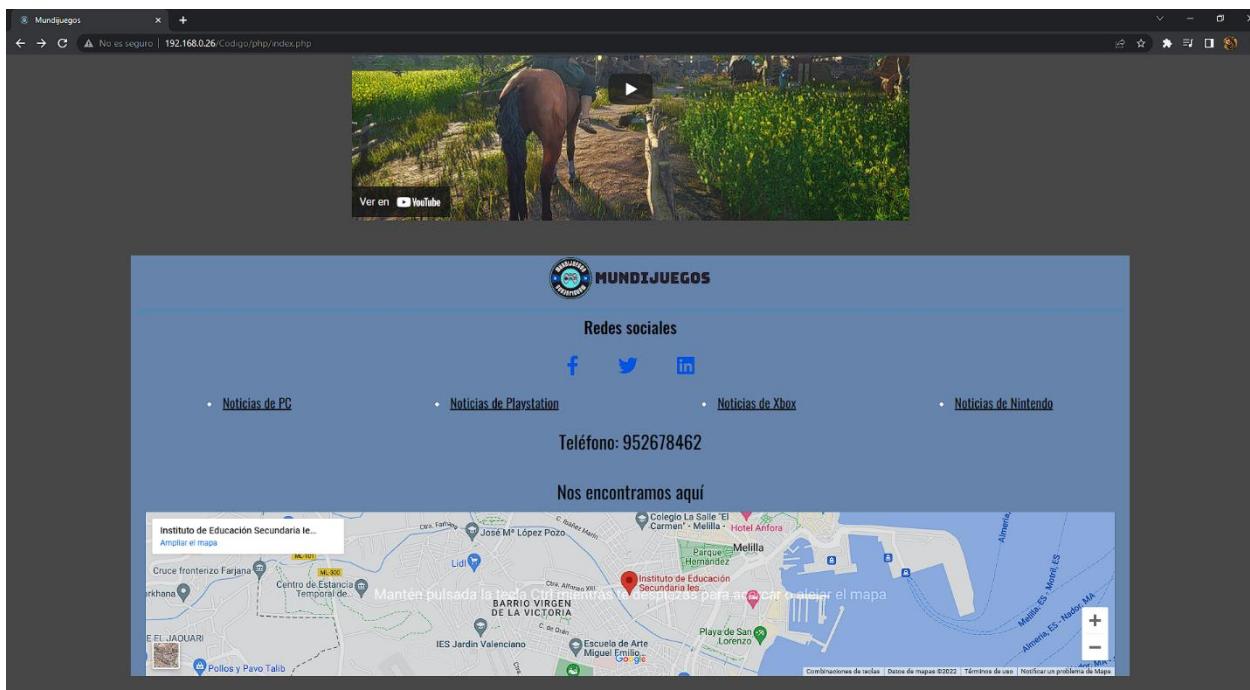
Se ha utilizado para crear el diseño de la página y el diseño responsive. Se ha utilizado la versión 3.4.

PRODUCTO

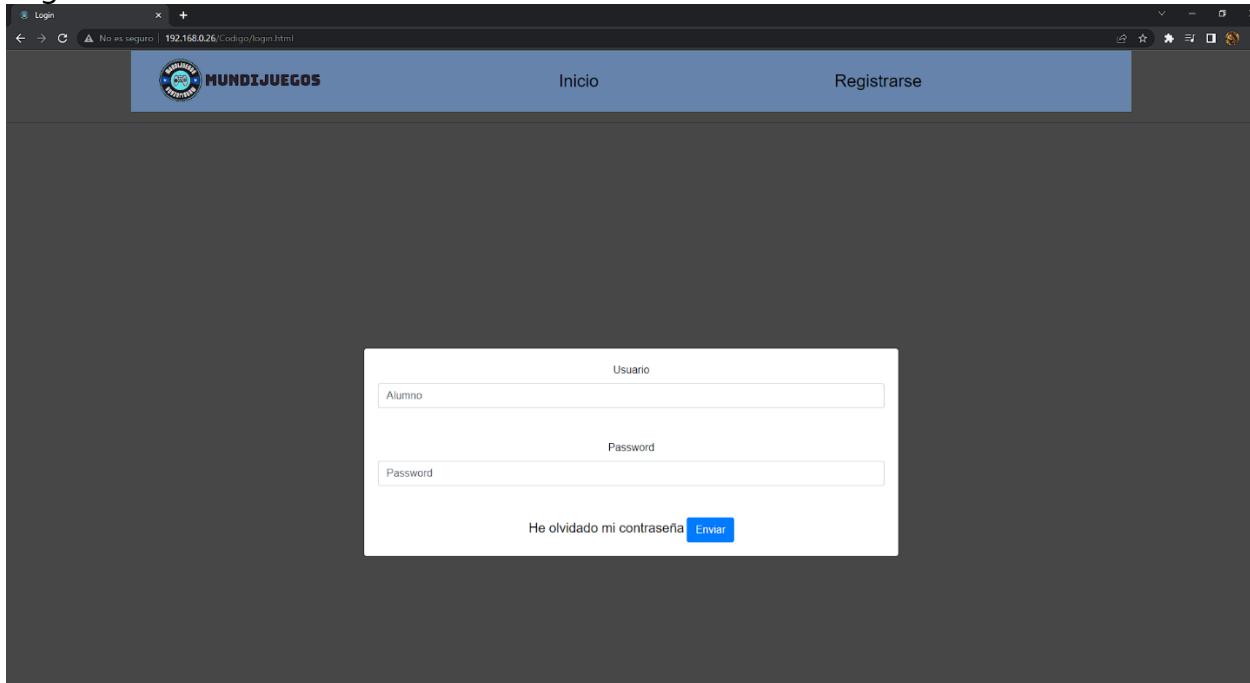
Página de Inicio sin un usuario iniciar sesión:

The screenshot shows the homepage of Mundijuegos. At the top, there is a navigation bar with links for 'Iniciar Sesión' (Login) and 'Registrarse' (Register), along with a search bar. The main content area features a dark background with a circuit board pattern. A welcome message reads: '¡BIENVENIDOS A MUNDIJUEGOS!' (Welcome to Mundijuegos!) and '¡TU PAGINA DE NOTICIAS!' (Your news page). Below this, a text block states: 'Aquí encontrarás todas las noticias y las últimas novedades en videojuegos. En Mundijuegos recopilamos cada día y minuto toda la actualidad del mundo de los videojuegos y sus diferentes plataformas para que siempre estés al corriente de la última actualización o contenido relacionado con tus títulos preferidos.' (Here you will find all the news and latest developments in video games. At Mundijuegos, we collect daily and minute-by-minute all the news from the world of video games and their different platforms so that you are always up-to-date with the latest update or content related to your favorite titles.) A section titled 'Plataformas en las que encontrarás noticias ↓' (Platforms where you will find news ↓) lists 'PC', 'Playstation', 'Xbox', and 'Nintendo'. Below this is a large image showing various Nintendo Switch models (Standard, OLED, and Lite) against a red background.

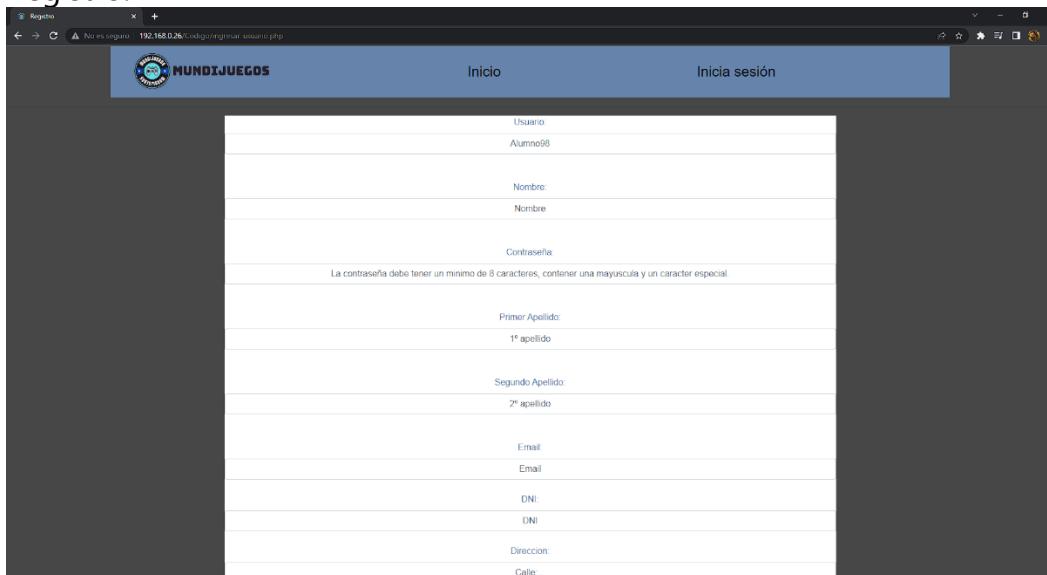
The screenshot shows the same Mundijuegos homepage as above, but with a different main content area. It features a video thumbnail for 'Video destacado de la semana' (Video of the week) with the subtitle '20 PRÓXIMOS JUEGOS PS4/PS5/XBOX ONE/SERIES X/PC PARA 2021' (20 upcoming games for PS4/PS5/Xbox One/Series X/PC for 2021). The video thumbnail shows a person riding a horse in a rural setting. Below the video, there is a section for 'Redes sociales' (Social media) with links for Facebook, Twitter, and LinkedIn. At the bottom of the page, there is a footer with links for 'Noticias de PC', 'Noticias de Playstation', 'Noticias de Xbox', and 'Noticias de Nintendo'. The footer also includes a phone number 'Teléfono: 952678462' and a map showing the location of the website's server.



Login:



Registro:



Registro

No es seguro | 192.168.0.27/Codigo/ingresar_usuario.php

Segundo Apellido:

2º apellido

Email:

Email

DNI:

DNI

Direccion:

Calle...

Fecha de nacimiento:

dd/mm/aaaa

Telefono:

Ejemplo: 659777894

Codigo Postal:

Ejemplo: 52005

Provincia: Elija una provincia

Municipio: Elija una opcion

Sección noticias PC:

Noticias PC

No es seguro | 192.168.0.27/Codigo/php/noticiasv.php

MUNDIJUEGOS

Atrás



Los juegos de League of Legends suman 180 millones de jugadores activos, más que los que tiene Steam.

PC

[Ver detalles](#)



Update on Our Pipeline

Diablo 4 y Overwatch 2 se harán esperar: Blizzard anuncia su retraso tras los cambios de personal.

PC

[Ver detalles](#)



Resident Evil 7 y Monster Hunter: World siguen destacando como los juegos más exitosos de Capcom.

PC

[Ver detalles](#)







Sección noticias Playstation:

Noticias Playstation

No es seguro | 192.168.0.27/Codigo/php/noticiasplay.php

MUNDIJUEGOS

Atrás



El gameplay de Elden Ring tiene todo lo que necesitas saber del juego: acción, mundo abierto y dragones.

PlayStation5

[Ver detalles](#)



Marvel's Spider-Man podría haber sido diferente: Insomniac cambió el final para evitar el crunch.

PlayStation5

[Ver detalles](#)



Impresiones Forspoken, una prometedora y espectacular aventura de rol y acción en mundo abierto.

PlayStation5

[Ver detalles](#)



Sniper Elite 5 es preciso y letal en su tráiler de presentación: el shooter confirma ventana de lanzamiento.

PlayStation5



Nova Pink, Starlight Blue y Galactic Purple son los nuevos DualSense para PlayStation 5.

PlayStation5

Sección noticias XBOX:

The screenshot shows a news section for Xbox. It features three news cards with the following details:

- Comparan Forza Horizon 5 en sus versiones para Xbox One S/X, Xbox Series S/X y PC.** (Xbox) [Ver detalles](#)
- Xbox All Access llega a España, financia tu Xbox Series X|S junto a 24 meses de Game Pass.** (Xbox) [Ver detalles](#)
- El stock de Xbox Series X mejorará a tiempo para estas navidades.** (Xbox) [Ver detalles](#)

Sección noticias Nintendo:

The screenshot shows a news section for Nintendo. It features three news cards with the following details:

- Nintendo Switch se acerca a Wii: alcanza 92,87 millones de consolas vendidas.** (Nintendo) [Ver detalles](#)
- Nintendo Switch logra el mejor mes de noviembre a nivel de ventas en España desde su lanzamiento.** (Nintendo) [Ver detalles](#)
- Sonic Frontiers sorprende con su aspecto visual y mundo abierto en el tráiler del nuevo juego de la saga.** (Nintendo) [Ver detalles](#)

Below these cards, there is another news card:

- Los remake de Pokémon Diamante y Perla continúan su racha en Japón y triunfan una semana más.** (Nintendo) [Ver detalles](#)

Página de los detalles de una noticia sin el usuario iniciar sesión:

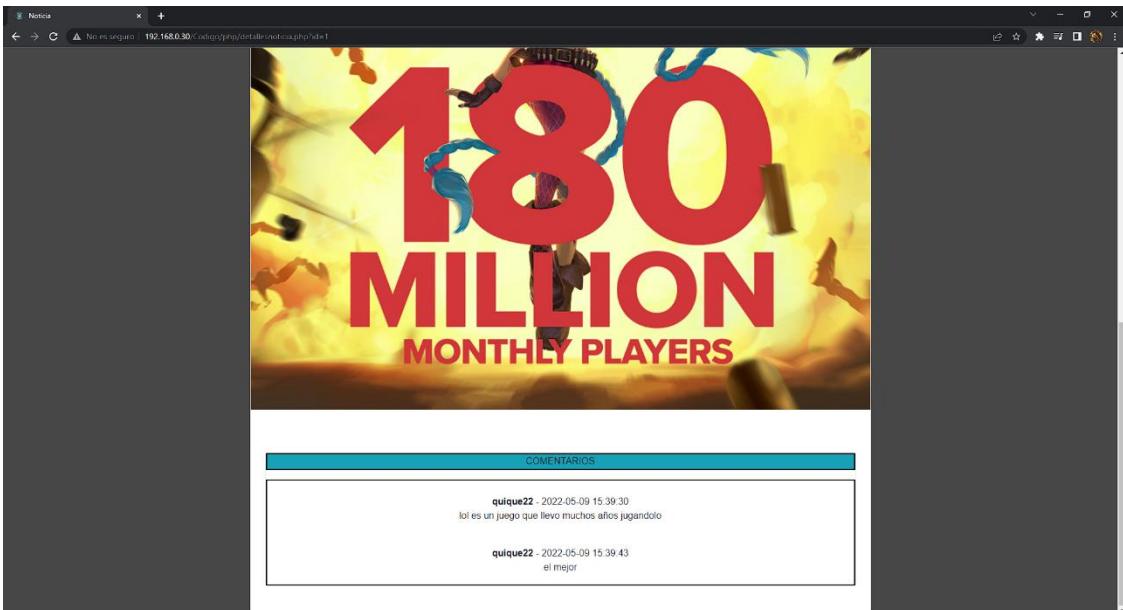
The screenshot shows a detail page for a news article about League of Legends. The headline is:

Los juegos de League of Legends suman 180 millones de jugadores activos, más que los que tiene Steam.

The main text of the article reads:

Es uno de los videojuegos más influyentes e importantes de la historia, pero que consiguiera ese título hace ya unos años no significa que esté de capa caída. Ni mucho menos: League of Legends es un fenómeno muy en forma, y el resto de juegos (y no solo juegos) pertenecientes a su universo funcionan a buen ritmo también. Y si, por alguna razón, esperabáis noticias diferentes al respecto, Riot Games ha dejado claro que las cosas siguen y

Inicia sesión para poder ver la noticia al completo y poder comentar!



Página de inicio de un usuario registrado:

MUNDIJUEGOS

Bienvenidos a MUNDIJUEGOS!
TU PÁGINA DE NOTICIAS!

Aquí encontrarás todas las noticias y las últimas novedades en videojuegos.
En Mundijuegos recopilamos cada día y minuto a minuto toda la actualidad del mundo de los videojuegos y sus diferentes plataformas para que siempre estés al corriente de la última actualización o contenido relacionado con tus títulos preferidos.

↓ Plataformas en las que encontrarás noticias ↓

- PC
- Playstation
- Xbox
- Nintendo

↓ ¡Noticias con más comentarios! ↓

- Los juegos de League of Legends suman 180 millones de jugadores activos, más los que tiene Steam.
- Nintendo Switch logra el mejor mes de noviembre a nivel de ventas en España desde su lanzamiento.
- Impresiones Ferispikes, una prometedora y espectacular aventura de rol y acción en mundo abierto.
- El desarrollo de GTA VI pospone el año "cotizado" y desaparece de los fans si se anuncia pronto, según un lector.
- Starfield Reporta con detalle en español Bethesda confirma en qué idiomas se lanzará el RPG espacial.
- Neon Pink, Starlight Blue y Galactic Purple son los nuevos DualSense para PlayStation 5.

Video destacado de la semana |

↓ Video destacado de la semana ↓

20 PRÓXIMOS JUEGOS PS4/PS5/XBOX ONE/SERIES X/PC PARA 2021

MUNDIJUEGOS

Redes sociales

f t in

Noticias de PC Noticias de Playstation Noticias de Xbox Noticias de Nintendo

Teléfono: 952678462

Nos encontramos aquí

Perfil:

Usuario	Password	Nombre	Primer Apellido	Segundo Apellido	Telefono	DNI	Email	Direccion	CP	Provincia	Municipio	Fecha de Nacimiento	
quique22	quiquequique	quique	perez	perez	693000683	453228955	quique@gmail.com	Calle Coral Edificio Ramos Número 23	52006	47	Becilla	1998-05-26	Eliminar usuario Modificar

Detalles de una noticia con el usuario iniciado:

Los juegos de League of Legends suman 180 millones de jugadores activos, más que los que tiene Steam.

Es uno de los videojuegos más influyentes e importantes de la historia, pero que consiguiera ese título hace ya unos años no significa que esté de capa caída. Ni mucho menos: League of Legends es un fenómeno muy en forma, y el resto de juegos (y no solo juegos) pertenecientes a su universo funcionan a buen ritmo también. Y si, por alguna razón, esperábamos noticias diferentes al respecto, Riot Games ha dejado claro que las cosas siguen yendo bien. En PCGamer han preguntado acerca de los títulos en concreto que forman este número, y parece abarcar los jugadores de League of Legends, League of Legends: Wild Rift, Teamfight Tactics y Battle of Golden Spatula (en China), por lo que no se está contando, por ejemplo, Valorant.

180 MILLION MONTHLY PLAYERS

Escribe un comentario sobre la noticia

Enviar comentario

COMENTARIOS

quique22 - 2022-05-09 15:39:30
lol es un juego que llevo muchos años jugandolo

quique22 - 2022-05-09 15:39:43
el mejor

Página principal de un usuario con el rol de Administrador:

Mundijuegos

No es seguro 192.168.0.30/Codigo/php/index.php

MUNDIJUEGOS

Cerrar Sesión Buscar

Panel de administración

Users **Plataformas** **Noticias** **Comentarios**

Bienvenidos a MUNDIJUEGOS!
TU PAGINA DE NOTICIAS!

Aquí encontrarás todas las noticias y las últimas novedades en videojuegos.
En Mundijuegos recopilamos cada día y minuto a minuto toda la actualidad del mundo de los videojuegos y sus diferentes plataformas para que siempre estés al corriente de la última actualización o contenido relacionado con tus títulos preferidos.

Plataformas en las que encontrarás noticias

• PC • Playstation • Xbox • Nintendo

Noticias con más comentarios!

Los juegos de League of Legends superan 180 millones de jugadores activos, más que los que tiene Steam.
[Ver más >](#)

Nintendo Switch logra el mejor mes de noviembre a nivel de ventas en España desde su lanzamiento.
[Ver más >](#)

Impresiones Forgeron, una prometedora y espectacular aventura de rol y acción en mundo abierto.
[Ver más >](#)

El desarrollo de GTA VI podría estar siendo "caótico" y desorganizado a los fans si se anuncia premio, según un insider.
[Ver más >](#)

Starfield llegará con diseño espacial. Bethesda confirma en qué瞬间 se mostrará el RPG espacial.
[Ver más >](#)

Nova Pink, Starlight Blue y Galactic Purple son los nuevos DualSense para PlayStation 5.
[Ver más >](#)

Noticias con más comentarios!

20 PRÓXIMOS JUEGOS PS4/PS5/XBOX ONE/SERIES X/PC PARA 2021
Ver en YouTube

Video destacado de la semana

20 PRÓXIMOS JUEGOS PS4/PS5/XBOX ONE/SERIES X/PC PARA 2021

Ver en YouTube

MUNDIJUEGOS

Redes sociales

[Facebook](#) [Twitter](#) [LinkedIn](#)

Noticias de PC Noticias de Playstation Noticias de Xbox Noticias de Nintendo

Teléfono: 952678462

Nos encontramos aquí

Página Usuarios:

Panel de Usuarios

No es seguro | 192.168.0.30/Codigo/php/panel.php

MUNDIJUEGOS

Atrás

ID Usuario	Usuario	Password	Nombre	Apellido1	Apellido2	Teléfono	Email	CP	Provincia	Comunidad Autónoma	Dirección	Rol	DNI	Fecha Nacimiento		
1	ainhoa	ainhoa	ainhoa	perez	zapata	698854478	ainhoa@gmail.com	52006	52	Melilla	nose	admin	45787878V	15/04/1994	Cambiar Rol	Eliminar
2	ainhoapz	ainhoapz	ainhoa	perez	zapata	630776293	ainhoaperez@gmail.com	52006	52	Melilla	calle lopezl	usuario	45342789D	11/02/1999	Cambiar Rol	Eliminar
77	quique22	quiquequique	quique	perez	perez	693000683	quique@gmail.com	52006	47	Becilla	Calle Coral Edificio Ramos Número 23	usuario	45322895S	1998-05-26	Cambiar Rol	Eliminar

Cambiar Rol de un usuario:

Modificar Usuario

No es seguro | 192.168.0.30/Codigo/php/modificarusuario.php?idUsuario=%20

MUNDIJUEGOS

Atrás

Role de Usuario ▾

Actualizar

Plataformas:

Plataformas Administrador

No es seguro | 192.168.0.30/Codigo/php/adminplataforma.php

MUNDIJUEGOS

Atrás

Insertar nueva consola

ID Plataforma	Nombre	Imagen		
5	Playstation		Modificar	Eliminar
7	Nintendo		Modificar	Eliminar
10	PC		Modificar	Eliminar
14	Xbox		Modificar	Eliminar

Modificar Consolas:

MUNDIJUEGOS

Atrás

Nombre Plataforma
Imagen
Seleccionar archivo Ninguno archivo selec.

Modificar

Insertar Consola:

MUNDIJUEGOS

Atrás

Nombre
Imagen
Seleccionar archivo Ninguno archivo selec.

Enviar

Noticias:

MUNDIJUEGOS

Atrás

Insertar nueva noticia

Título	Descripción	Imagen	Plataforma	Modificar	Eliminar
Los juegos de League of Legends suman 180 millones de jugadores activos, más que los que tiene Steam.	Es uno de los videojuegos más influyentes e importantes de la historia, pero que consiguiera ese título hace ya unos años no significa que esté de capa caída. Ni mucho menos: League of Legends es un fenómeno muy en forma, y el resto de juegos (y no solo juegos) pertenecientes a su universo funcionan a buen ritmo también. Y si, por alguna razón, esperábamos noticias diferentes al respecto, Riot Games ha dejado claro que las cosas siguen yendo bien. En PCGamer han preguntado acerca de los títulos en concreto que forman este número, y parece abarcar los jugadores de League of Legends, League of Legends: Wild Rift, Teamfight Tactics y Battle of Golden Spatula (en China), por lo que no se está contando, por ejemplo, Valorant.		PC	Modificar	Eliminar
Diablo 4 y Overwatch 2 se harán esperar: Blizzard anuncia su retraso tras los cambios de personal.	La crisis institucional de Activision Blizzard ha supuesto un terremoto en la industria tanto dentro como fuera de la compañía. Una parte de la plantilla se ha visto afectada directa e indirectamente por los casos de acoso y las denuncias, y piezas clave en el desarrollo de proyectos como Diablo IV se han visto fuera de la compañía y sustituidos por nuevos creativos.		PC	Modificar	Eliminar
Resident Evil 7 y Monster Hunter: World siguen destacando como los juegos más exitosos de Capcom.	Capcom sigue copando la actualidad del videojuego gracias a sus últimos números, pues han superado las expectativas de todo el mundo gracias a los éxitos de Monster Hunter y Resident Evil. Ya que, con un Resident Evil 7 que no deja de acumular beneficios y un Monster Hunter: World que ya lidera la lista de la compañía, Capcom ha demostrado que todavía no ha tocado techo.		PC	Modificar	Eliminar
El gameplay de Elden Ring tiene todo lo que necesitas saber del juego: acción, mundo abierto y dragones.	Han tardado en dar novedades pero iba aya novedades! El equipo japonés de FromSoftware ha presentado un extenso video gameplay de Elden Ring que nos ha permitido ver en acción su esperadísimo RPG de mundo abierto: además de confirmar que habrá una edición especial para coleccionistas (que se filtró minutos antes) con una figura y hasta una réplica del casco de Malenia, uno de los personajes principales de la aventura. Este largo gameplay ha arrancado con el combate contra un dragón, aunque aquí la novedad radica en el hecho de que nuestro protagonista va en una montura, que es una de las nuevas características que introduce Elden Ring con respecto a otros juegos de la saga Souls. Durante el gameplay se han mostrado otros detalles interesantes de este nuevo universo de fantasía, con un gran y bello mapa ilustrado que nos va a permitir explorar con mayor comodidad el vasto y peligroso mundo abierto del que hace gala el juego de FromSoftware. Durante nuestro viaje por este nuevo mundo podemos convocar a otros jugadores para disfrutar de una renovada experiencia multijugador, haciendo frente no solo a los jefes finales sino también a los propios peligros de este universo de fantasía. Con la ayuda de las monturas podemos recorrer largas distancias en menor tiempo y, en la línea del fantástico Sekiro, la verticalidad de los escenarios nos permitirá usar el sigilo y los ataques furtivos para derrotar a los enemigos antes de que se den cuenta. Por supuesto, siendo un juego de FromSoftware, además de ese gran mundo abierto también encontraremos mazmorras, cuevas,		PlayStation5	Modificar	Eliminar

Modificar Noticias:

A screenshot of a web browser window titled "Modificar Noticia". The URL is "No es seguro | 192.168.0.30/Codigos/php/modificarNoticia.php?idNoticias=%201". The page has a blue header with the "MUNDIJUEGOS" logo. Below the header, there is a "Atrás" link. The main content area contains several input fields: "Título" (Title), "Descripción" (Description), "Imagen" (Image) with a "Seleccionar archivo" (Select file) button and a message "Ninguno archivo selec.", "Plataforma" (Platform), "Nombre Plataforma" (Name Platform), and a "Modificar" (Modify) button.

Insertar nueva noticia:

A screenshot of a web browser window titled "Insertar Noticia". The URL is "No es seguro | 192.168.0.30/Codigos/insertarNoticia.html". The page has a blue header with the "MUNDIJUEGOS" logo. Below the header, there is a "Atrás" link. The main content area contains several input fields: "Título" (Title), "Número Plataforma" (Platform Number), "Nombre Plataforma" (Name Platform), "Descripción" (Description), "Imagen" (Image) with a "Seleccionar archivo" (Select file) button and a message "Ninguno archivo selec.", and an "Insertar" (Insert) button.

Comentarios:

The screenshot shows a table of comments from a website. The columns are labeled: idComentarios, idNoticias, idUsuario, Contenido, and Estado. The comments are as follows:

idComentarios	idNoticias	idUsuario	Contenido	Estado	Aceptar	Eliminar
54	1	77	lol es un juego que llevo muchos años jugandolo	Revision	Aceptar	Eliminar
55	1	77	el mejor	Revision	Aceptar	Eliminar
56	27	77	esperado ansioso a que salga!	Revision	Aceptar	Eliminar
57	24	77	la mejor consola que hay	Revision	Aceptar	Eliminar
58	29	77	que ganas de que salga yaaa	Revision	Aceptar	Eliminar
59	32	77	el color azul mola mucho	Revision	Aceptar	Eliminar
60	30	77	juegazooo	Revision	Aceptar	Eliminar

Olvido de contraseña:

The screenshot shows a password recovery form. It has fields for 'Usuario' (with value 'Alumno98') and 'Email' (with value 'Email'). There is also a blue 'Enviar' button.

Usuario	Alumno98
Email	Email
<input type="button" value="Enviar"/>	

BIBLIOGRAFÍA

Comentarios:

<https://www.youtube.com/watch?v=EHca8OI8ez0>

Carrusel Glider:

<https://nickpiscitelli.github.io/Glider.js/>

<https://www.youtube.com/watch?v=uDqW8bCFhps>

Tarjetas bootstrap:

<https://getbootstrap.com/docs/4.0/components/card/>

Imágenes en html:

https://developer.mozilla.org/es/docs/Learn/HTML/Multimedia_and_embedding/Images_in_HTML

Estilos tablas bootstrap:

<https://getbootstrap.com/docs/5.0/content/tables/>

Texto:

<https://www.tutorialmonsters.com/curso-php-como-cortar-una-cadena-de-texto/>

<https://www.srccodigofuente.es/tutoriales/ver-tutorial/como-cortar-texto-en-php>

