29/05/2017



GamersHUB, noticias de videojuegos. Desarrollo con Laravel y despliegue en Amazon Web Services

C.I.F.P. JUAN de colonia | C.F.G.S. DESARROLLO DE APLICACIONES WEB

Braulio Álvaro Peso García

PROYECTO de desarrollo de aplicaciones web

Índice de Contenidos

[1.Introducción 3](#__RefHeading___Toc1358_1738586612)

[1.1Descripción 3](#__RefHeading___Toc1360_1738586612)

[1.2Justificación 3](#__RefHeading___Toc1362_1738586612)

[1.3Glosario de términos 4](#__RefHeading___Toc1364_1738586612)

[2.Planificación 5](#__RefHeading___Toc1366_1738586612)

[2.1Requisitos funcionales y no funcionales 5](#__RefHeading___Toc1368_1738586612)

[2.2Recursos 7](#__RefHeading___Toc1370_1738586612)

[2.3Planificación temporal 8](#__RefHeading___Toc1372_1738586612)

[2.4Planificación económica 9](#__RefHeading___Toc1374_1738586612)

[3.Tecnologías 9](#__RefHeading___Toc1376_1738586612)

[4.Desarrollo 9](#__RefHeading___Toc1378_1738586612)

[4.1Diseño 9](#__RefHeading___Toc1380_1738586612)

[4.2Pruebas 9](#__RefHeading___Toc1382_1738586612)

[5.Conclusiones finales 9](#__RefHeading___Toc1384_1738586612)

[5.1Grado de cumplimiento de los requisitos fijados y análisis de las desviaciones 9](#__RefHeading___Toc1386_1738586612)

[5.2Propuestas de mejora o ampliaciones futuras 9](#__RefHeading___Toc1388_1738586612)

[6.Guías 9](#__RefHeading___Toc1390_1738586612)

[6.1Guía del usuario 9](#__RefHeading___Toc1392_1738586612)

[6.2Guía del administrador 9](#__RefHeading___Toc1394_1738586612)

[7.Referencias bibliográficas 9](#__RefHeading___Toc1396_1738586612)

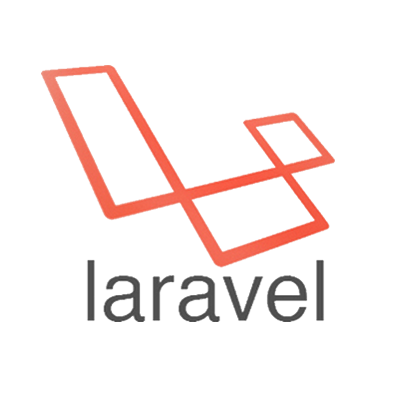
# Introducción

## Descripción

Este proyecto ha sido realizado con la finalidad de ofrecer un portal dirigido a usuarios que jueguen a videojuegos y puedan mantenerse informados de las últimas novedades en este ámbito. Además de poder acceder a diferentes servicios que se ofrecen dentro de la aplicación: foros, grupos de usuarios, conexión con diferentes plataformas de videojuegos.

La aplicación será desarrollada en PHP 7 mediante el framework Laravel y haciendo uso de tecnologías como HTML5, CSS3, AJAX, JavaScript, jQuery. Estructurándose todo esto en una aplicación MVC mediante el motor de plantillas Blade y el ORM Eloquent.

El desarrollo será realizado con las herramientas de IntelliJ: PhpStorm, WebStorm y DataGrip. Además utilizaré el sistema de control de versiones Git a través de la plataforma GitHub mediante un repositorio privado.

La aplicación será desplegada posteriormente gracias a la colección de de servicios de computación en la nube ofrecidos a través de internet por AWS.

## Justificación

He decidido realizar estre proyecto debido a mi afición a los videojuegos y como jugador por tener la necesidad de tener un portal donde consultar de manera eficaz las últimas novedades de videojuegos, información sobre videojuegos y sobre todo de compartir con otros usuarios mi misma afición.

También como oportunidad de aplicar nuevas tecnologías que he aprendido por mi cuenta además de las aprendidas durante el curso de Desarrollo de Aplicaciones Web de forma que éstos conocimientos me sean de utilidad en el futuro.

# Planificación

## Requisitos funcionales y no funcionales

| **REQUISITOS FUNCIONALES** | |
| --- | --- |
| **REQUERIMIENTO** | **DESCRIPCIÓN** |
| **RF1** | La aplicación contará con un registro de usuarios. |
| **RF2** | Cada usuario contará con perfil público donde se mostrará la actividad reciente en la aplicación. |
| **RF3** | El usuario podrá determinar la privacidad de ciertos elementos de su perfil de usuario público. |
| **RF4** | El usuario podrá realizar publicaciones públicas en un muro personal visible en su perfil. |
| **RF5** | El usuario podrá crear grupos privados y/o públicos y determinar la configuración de estos. |
| **RF6** | El usuario podrá acceder a los distintos grupos públicos y/o privados existentes en la aplicación. |
| **RF7** | El usuario podrá añadir como amigo/a a otros usuarios registrados en la aplicación. |
| **RF8** | El usuario podrá enviar mensajes privados a otros usuarios registrados en la aplicación. |
| **RF9** | El usuario podrá reportar a otros usuarios si sospechan que han cometido alguna infracción. |
| **RF10** | El usuario podrá crear nuevos hilos en los foros y responder a éstos. |
| **RF11** | El usuario podrá comentar en los artículos. |
| **RF12** | La aplicación contendrá un back-end de administración para gestionar: artículos, usuarios, categorías, foros, imágenes y la configuración general de la aplicación. |
| **RF13** | Los artículos podrán ser de tres tipos: noticia, avance, vídeo o análisis. |
| **RF14** | Si el artículo es de tipo análisis se deberá especificar las siguientes notas: apartado visual, apartado sonoro, apartado técnico, apartado de jugabilidad. Y una conclusión final. |
| **RF15** | Se podrá expulsar a determinados usuarios e impedirles la entrada a la aplicación por: email o IP pública. |
| **RF16** | La aplicación almacenará las imágenes en un servidor de almacenamiento flexible ofrecido por AWS. |
| **RF17** | La aplicación contará con una búsqueda avanzada para encontrar: artículos, juegos, usuarios o grupos. |
| **RF18** | La aplicación contará con un sistema de roles y permisos para restringir acciones a los usuarios registrados y dividir las funciones del equipo administrativo (administrador, moderador, redactor...) |
| **RF19** | La aplicación contará con un feed RSS Atom con los últimos 20 artículos redactados. |
| **RF20** | El registro contará con un código de seguridad (reCaptcha). |
| **RF21** | Cuando un usuario se registre, éste deberá confirmar su dirección de correo electrónico enviando a éste un mail de confirmación. |
| **RF22** | Se notificará al usuario de determinadas acciones o procesos mediante el envío de correos electrónicos a su bandeja de correo (respuestas en comentarios o temas, solicitudes de amigo, mensajes privados). |
| **RF23** | Las categorías podrán ser plataformas relacionandose éstas con algún elemento hardware (consolas de videojuegos, equipos informáticos...). |
| **RF24** | Los usuarios se podrán registrar en la aplicación mediante su cuenta de Facebook, Twitter, Google u otras redes sociales. |
| **RF25** | Cuando se publique un artículo, se deberá conectar con la API de Twitter para enviar un tweet a una cuenta determinada. |
| **RF26** | La aplicación contactará con diferentes API’s para obtener información sobre juegos. |
| **REQUISITOS NO FUNCIONALES** | |
| **RNF1** | La aplicación debe ser fácil de utilizar. |
| **RNF2** | Las interfaces de la aplicación deben ser amigables e intuitivas. |
| **RNF3** | La aplicación debe de proporcionar tiempos de respuesta rápidos. |
| **RNF4** | La aplicación debe ser fácil de analizar y modificar para corregir posibles fallos. |
| **RNF5** | El código fuente de la aplicación deberá estar correctamente documentado. |
| **RNF6** | La aplicación debe de mantener los datos almacenados seguros y protegidos. |

## Recursos

| **RECURSOS HARDWARE** | |
| --- | --- |
| **RECURSO** | **DESCRIPCIÓN** |
| **Ordenador** | Equipo personal para desarrollar el proyecto. |
| **Servidor Amazon Elastic Compute Cloud (AWS EC2)** | Entorno virtual de cómputo que permite utilizar interfaces de servicios web para lanzar instancias con distintos sistemas operativos, cargarlas con su entorno de aplicaciones personalizado, administrar sus permisos de acceso a la red y ejecutar su imagen utilizando diferentes sistemas. |
| **Servidor de almacenamiento flexible (AWS S3)** | Servidor de almacenamiento de objetos para almacenar y recuperar la gran cantidad de datos desde cualquier ubicación de la web. |
| **Servicio DNS (AWS Route 53)** | Servicio web DNS escalable y de alta disponibilidad en la nube. |
| **Servidor de base de datos relacionales (AWS RDS)** | Servicio administrado de base de datos relacional (Amazon Aurora, MySQL, MariaDB, Oracle, Microsoft SQL Server y PostgreSQL) |
| **Servicio de despliegue (AWS Elastic Beanstalk)** | Servicio para implementar y escalar servicios y aplicaciones web desarrollados con Java, .NET, PHP, Node.js, Python, Ruby, Go y Docker en servidores familiares como Apache, Nginx, Passenger e IIS. |
| **Servicio SMTP (AWS SES)** | Servicio de envío de correo electrónico. |
| **RECURSOS SOFTWARE** | |
| **JetBrains WebStorm** | Entorno de desarrollo para JavaScript, HTML y CSS |
| **JetBrains PhpStorm** | Entrono de desarrollo para PHP |
| **JetBrains DataGrip** | Sistema gestor de bases de datos |
| **Adobe Photoshop CC** | Editor fotográfico. |
| **Microsoft Office Word y PowerPoint** | Procesador de textos y editor de presentaciones. |
| **GIT** | Sistema de control de versiones. |

## Planificación temporal

|  |  |
| --- | --- |
| **FECHA** | **ACTIVIDAD** |
| **6-marzo** | Inicio del proyecto. Establecer los requisitos del sistema y del software. |
| **8-marzo** | Creación de la estructura de datos relacional y de la arquitectura interna de la aplicación: mapeo de modelos, MVC, implementación de librerías externas necesarias e instalación del framework Laravel.  Diseño preliminar de las interfaces de usuario.  Documentación de lo realizado en la memoria del proyecto. |
| **19-marzo** | Maquetación de las interfaces de usuario e implementación de las mismas en la aplicación.  Documentación de lo realizado en la memoria del proyecto. |
| **29-marzo** | Codificación del sistema de usuarios. |
| **6-abril** | Codificación del sistema de artículos. |
| **13-abril** | Codificación del sistema de foros y grupos. |
| **27-abril** | Codificación del back-end e implementación de API’s REST para la creación de elementos en la base de datos. |
| **11-mayo** | Depuración del código y realización de pruebas de la lógica interna de la aplicación y de funciones externas.  Documentación de lo realizado en la memoria del proyecto. |
| **18-mayo** | Exportar la aplicación del entorno de desarrollo para su posterior despliegue. |
| **21-mayo** | Despliegue y configuración de la aplicación en la nube a través de los servicios ofrecidos por Amazon Web Services. |
| **22-mayo en adelante** | Explotación y mantenimiento de la aplicación. |

## Planificación económica

|  |  |  |
| --- | --- | --- |
| **DESARROLLO DE LA APLICACIÓN** | | |
| **HORAS INVERTIDAS** | **PRECIO/HORA** | |
| 250 horas | 25€/hora | |
| **GASTOS EN SOFTWARE** | | |
| **RECURSO** | | **PRECIO** |
| Licencias de uso para el software de desarrollo. | | 35,00€/mes |
| **GASTOS EN HARDWARE** | | |
| **RECURSO** | | **PRECIO** |
| Amortización mensual del hardware. | | 50,00€/mes |
| **GASTOS EN EL DESPLIEGUE DE LA APLICACIÓN** | | |
| **RECURSO** | | **PRECIO** |
| Servidor Amazon Elastic Compute Cloud (AWS EC2) | | ≈28€/mes |
| Servidor de almacenamiento flexible (AWS S3) | | ≈15€/MES + 12€/1000 peticiones (PUT,GET) |
| Servidor de base de datos relacionales (AWS RDS) | | ≈15€/mes |
| Servicio SMTP (AWS SES) | | ≈5€ c/1000 peticiones |
| Dominio www.gamershub.es | | 10€/año |

|  |  |
| --- | --- |
| **TOTAL GASTOS EN SOFTWARE + HARDWARE** | 85,00€/mes |
| **TOTAL DESPLIEGUE + MANTENIMIENTO** | ≈151,20€/mes |
| **TOTAL DESARROLLO DE LA APLICACIÓN** | 6.250,00€ |

## Tecnologías

### PHP v7.0.15

Es lo que se denomina un lenguaje de programación del lado de servidor, se utiliza principalmente para la creación de páginas webs dinámicas. El servidor procesa las peticiones PHP previamente embebidas en el código HTML y devuelve una página web.

### HTML5

HTML es un lenguaje de marcado que se utiliza para el desarrollo de páginas de Internet. Se trata de la sigla que corresponde a HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto, que podría ser traducido como Lenguaje de Formato de Documentos para Hipertexto.

HTML5 es la última versión de HTML. El término representa dos conceptos diferentes: Se trata de una nueva versión de HTML, con nuevos elementos, atributos y comportamientos. Contiene un conjunto más amplio de tecnologías que permite a los sitios Web y a las aplicaciones ser más diversas y de gran alcance.

### CSS3

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.

### JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.

### AJAX

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página.

### jQuery 3.2.0

jQuery es una biblioteca multiplataforma de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

### Laravel Framework 5.3

Laravel es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5 y PHP 7. Su filosofía es desarrollar código PHP de forma elegante y simple, evitando el "código espagueti". Fue creado en 2011 y tiene una gran influencia de frameworks como Ruby on Rails, Sinatra y ASP.NET MVC.

### Blade

Laravel usa unos archivos que se llaman plantillas o templates que suelen ser nuestros archivos principales, que tienen los segmentos de código que se repiten en mas de una vista, como por ejemplo la barra de navegacion, un menú de opciones, la estructura del acomodo de nuestro proyecto, etc.

Además de los templates, se cuentan con archivos que se llaman partials, estos archivos son pequeños segmentos de código que suelen ser usados comunmente en partes del sistema en especifico.

Esto hace que las vistas de cada parte del proyecto, que suelen ser llamadas por una ruta o controlador sean mucho mas pequeñas que usando otro tipo de frameworks para desarrollo Web, y para poder unir todos estos archivos o piezas del rompecabezas usamos el motor de plantillas de Laravel llamado BLADE.

### Eloquent ORM

Object-Relational mapping, o lo que es lo mismo, mapeo de objeto-relacional, es un modelo de programación que consiste en la transformación de las tablas de una base de datos, en una serie de entidades que simplifiquen las tareas básicas de acceso a los datos para el programador.

Eloquent es el ORM que incluye Laravel para manejar de una forma fácil y sencilla los procesos correspondientes al manejo de bases de datos en nuestro proyecto, gracias a las funciones que provee podremos realizar complejas consultas y peticiones de base de datos sin escribir una sola línea de código SQL.

### Composer

Composer es un manejador de dependencias y librerías.

La instalación siempre es local para cualquier proyecto, las librerías se instalan en un directorio por defecto (normalmente es /vendor).

### REST APIs

La interfaz de programación de aplicaciones, abreviada como API del inglés: Application Programming Interface, es un conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

### MySQL

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual [GPL](https://es.wikipedia.org/wiki/Licencia_pública_general_de_GNU)/Licencia comercial por Oracle Corporation y está considerada como la base datos open source más popular del mundo, y una de las más populares en general junto a Oracle y [Microsoft SQL Server](https://es.wikipedia.org/wiki/Microsoft_SQL_Server), sobre todo para entornos de [desarrollo web](https://es.wikipedia.org/wiki/Desarrollo_web).

### Git

Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente.

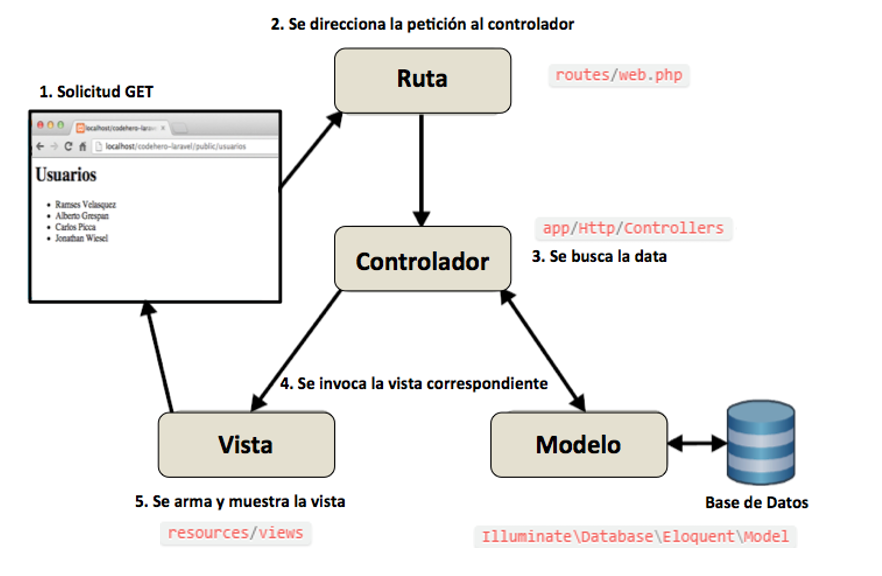
# 

# Desarrollo

## Diseño

## Diagrama de la aplicación

Laravel utiliza un sistema de rutas bajo un modelo vista controlador, en la siguiente imagen se muestra el proceso que se realiza cuando ingresamos a una URL además de la arquitectura del patrón MVC.



## Diagrama E/R, esquemas de la BBDD

### Diagrama de la base de datos

## Diccionario de datos

A continuación se muestra el diccionario de datos de la base de datos utilizada en la aplicación dividida por tablas.

**Articles**

En la siguiente tabla se almacenan todos los artículos los cuales pueden ser de cuatro tipos: Noticia, Análisis, Avance o Vídeo.

Estos artículos pueden estar relacionados con un juego determinado (game\_id) y han sido escritos por un usuario (user\_id).

|  |  |
| --- | --- |
| Campo | Descripción |
| id |  |
| user\_id |  |
| game\_id |  |
| type |  |
| image |  |
| title |  |
| description |  |
| content |  |
| seo\_optimized\_title |  |
| created\_at |  |
| updated\_at |  |

**Categories**

En la siguiente tabla se almacenan todas las categorías a las cuales puede pertenecer un artículo o un juego. Las categorías también pueden ser tratadas como plataformas dentro de la aplicación si están relacionadas con algún equipo de hardware como por ejemplo una videoconsola.

|  |  |
| --- | --- |
| Campo | Descripción |
| id |  |
| name |  |
| color |  |
| alias |  |
| Its\_platform |  |
| created\_at |  |
| updated\_at |  |

**Articles\_categories**

En la siguiente tabla se almacenan las relaciones entre artículos y categorías, es decir, las categorías a las que pertenece un artículo.

|  |  |
| --- | --- |
| Campo | Descripción |
| id |  |
| category\_id |  |
| article\_id |  |

**Articles\_comments**

En la siguiente tabla se almacenan los comentarios publicados por un usuario en un artículo.

|  |  |
| --- | --- |
| Campo | Descripción |
| id |  |
| article\_id |  |
| user\_id |  |
| comment |  |
| created\_at |  |
| updated\_at |  |

**Countries**

En la siguiente tabla se almacenan los diferentes países a los cuales puede pertenecer un usuario cuando se registra.

|  |  |
| --- | --- |
| Campo | Descripción |
| id |  |
| code |  |
| country |  |

**Developers**

En la siguiente tabla se almacenan los desarrolladores a los cuales puede pertenecer un juego

|  |  |
| --- | --- |
| Campo | Descripción |
| id |  |
| name |  |
| created\_at |  |
| updated\_at |  |

**Distributors**

En la siguiente tabla se almacenan los distribuidores a los cuales puede pertenecer un juego

|  |  |
| --- | --- |
| Campo | Descripción |
| id |  |
| name |  |
| created\_at |  |
| updated\_at |  |

**Events**

En la siguiente tabla se almacenan los diferentes eventos programados.

|  |  |
| --- | --- |
| Campo | Descripción |
| id |  |
| title |  |
| description |  |
| twitter\_hashtag |  |
| twitter\_widget\_id |  |
| related\_tag |  |
| start\_date |  |
| end\_date |  |
| seo\_optimized\_title |  |
| created\_at |  |
| updated\_at |  |

**Events\_events**

La siguiente tabla crea una relación entre diferentes eventos, pudiendo varios de ellos pertenecer a uno solo.

|  |  |
| --- | --- |
| Campo | Descripción |
| id |  |
| event\_id |  |
| child\_event\_id |  |
| created\_at |  |
| updated\_at |  |

**Forums**

La siguiente tabla almacena la información de los foros existentes.

|  |  |
| --- | --- |
| Campo | Descripción |
| id |  |
| title |  |
| type |  |
| forum\_section\_id |  |
| game\_id |  |
| category\_id |  |
| seo\_optimized\_title |  |
| created\_at |  |
| updated\_at |  |

**Forums\_section**

La siguiente tabla almacena las diferentes secciones en las cuales se puede categorizar un foro.

|  |  |
| --- | --- |
| Campo | Descripción |
| id |  |
| title |  |
| description |  |
| created\_at |  |
| updated\_at |  |

**Forums\_topics**

La siguiente tabla almacena los temas existentes en los diferentes foros.

|  |  |
| --- | --- |
| Campo | Descripción |
| id |  |
| title |  |
| content |  |
| type |  |
| forum\_id |  |
| user\_id |  |
| forum\_topic\_id |  |
| created\_at |  |
| updated\_at |  |

**Games**

La siguiente tabla almacena los diferentes juegos.

|  |  |
| --- | --- |
| Campo | Descripción |
| id |  |
| title |  |
| description |  |
| available\_on |  |
| players\_quantity |  |
| duration |  |
| language |  |
| release\_date |  |
| header\_image |  |
| boxed\_image |  |
| seo\_optimized\_title |  |
| created\_at |  |
| updated\_at |  |

**Games\_categories**

La siguiente tabla almacena las categorías a las cuales pertenecen los diferentes juegos.

|  |  |
| --- | --- |
| Campo | Descripción |
| id |  |
| game\_id |  |
| category\_id |  |

**Games\_genres**

La siguiente tabla almacena los géneros a los cuales pertenece un juego.

|  |  |
| --- | --- |
| Campo | Descripción |
| id |  |
| game\_id |  |
| genre\_id |  |

**Games\_developers**

La siguiente tabla almacena los desarrolladores a los cuales pertenece un juego.

|  |  |
| --- | --- |
| Campo | Descripción |
| id |  |
| game\_id |  |
| developer\_id |  |

**Games\_distributors**

La siguiente tabla almacena los distribuidores a los cuales pertenece un juego.

|  |  |
| --- | --- |
| Campo | Descripción |
| id |  |
| game\_id |  |
| genre\_id |  |

**Genres**

La siguiente tabla almacena los diferentes géneros de videojuegos.

|  |  |
| --- | --- |
| Campo | Descripción |
| id |  |
| name |  |
| created\_at |  |
| updated\_at |  |

**Permissions**

La siguiente tabla almacena los diferentes permisos que puede tener un usuario.

|  |  |
| --- | --- |
| Campo | Descripción |
| id |  |
| name |  |
| description |  |
| created\_at |  |
| updated\_at |  |

**Roles**

La siguiente tabla almacena los diferentes roles que puede tener un usuario.

|  |  |
| --- | --- |
| Campo | Descripción |
| id |  |
| name |  |
| description |  |
| created\_at |  |
| updated\_at |  |

**Roles\_permissions**

Asocia los diferentes permisos a un único rol, es decir, los agrupa.

|  |  |
| --- | --- |
| Campo | Descripción |
| id |  |
| name |  |
| description |  |
| created\_at |  |
| updated\_at |  |

**Reviews**

La siguiente tabla almacena los diferentes análisis de videojuegos relacionados con un artículo en concreto.

|  |  |
| --- | --- |
| Campo | Descripción |
| id |  |
| article\_id |  |
| game\_id |  |
| gameplay\_score |  |
| graphics\_score |  |
| sounds\_score |  |
| innovation\_score |  |
| created\_at |  |
| updated\_at |  |

**Tags**

La siguiente tabla almacena las diferentes etiquetas a las que puede pertenecer un artículo

|  |  |
| --- | --- |
| Campo | Descripción |
| id |  |
| article\_id |  |
| name |  |
| created\_at |  |
| updated\_at |  |

**Users\_email\_tokens**

La siguiente tabla almacena tokens de seguridad para confirmar una cuenta de usuario o reestablecer una contraseña.

|  |  |
| --- | --- |
| Campo | Descripción |
| id |  |
| user\_id |  |
| token |  |
| created\_at |  |
| updated\_at |  |

**Users\_privacy**

La siguiente tabla almacena la configuración de privacidad de los usuarios registrados.

|  |  |
| --- | --- |
| Campo | Descripción |
| id |  |
| user\_id |  |
| public\_profile |  |
| public\_city |  |
| public\_age |  |
| public\_accounts |  |
| public\_con\_accounts |  |
| created\_at |  |
| updated\_at |  |

**Videos**

La siguiente tabla almacena los diferentes videos que están relacionados a un artículo en concreto.

|  |  |
| --- | --- |
| Campo | Descripción |
| id |  |
| article\_id |  |
| youtube\_code |  |
| duration |  |
| views\_count |  |
| created\_at |  |
| updated\_at |  |

**Users**

La siguiente tabla almacena los usuarios registrados en la aplicación.

|  |  |
| --- | --- |
| Campo | Descripción |
| id |  |
| nickname |  |
| email |  |
| password |  |
| role\_id |  |
| name |  |
| surname |  |
| birthdate |  |
| country\_id |  |
| city |  |
| gender |  |
| avatar |  |
| favourite\_genre |  |
| signature |  |
| xbox\_gamertag |  |
| ps\_id |  |
| nintendo\_network |  |
| friend\_code\_wii |  |
| friend\_code\_3ds |  |
| friend\_code\_ds |  |
| microsoft\_gamertag |  |
| steam\_id |  |
| twitter |  |
| Facebook |  |
| google |  |
| web\_blog |  |
| remember\_token |  |
| verified |  |
| created\_at |  |
| updated\_at |  |

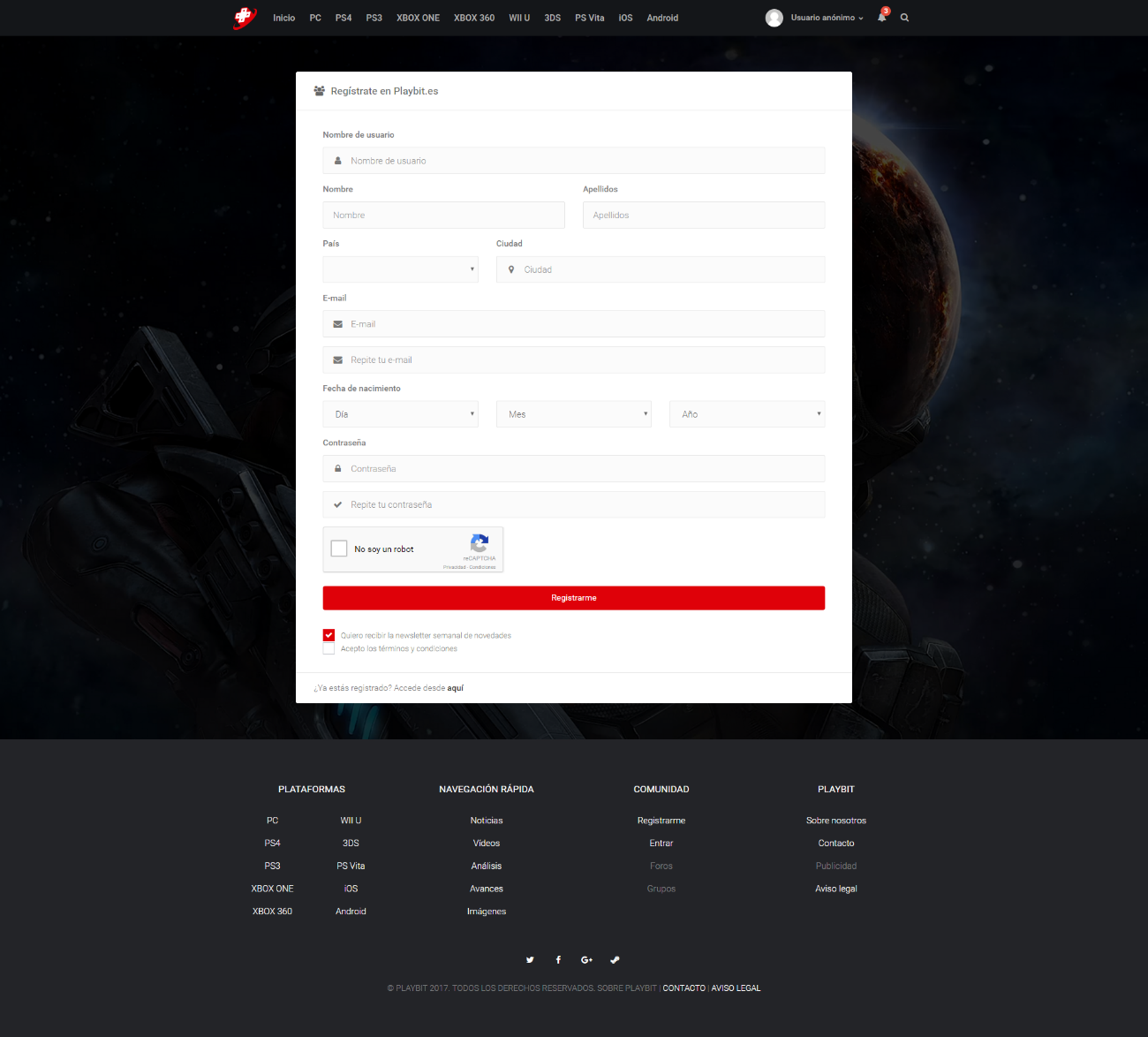
## Diseño de interfaces

**Página de inicio**

Descripción

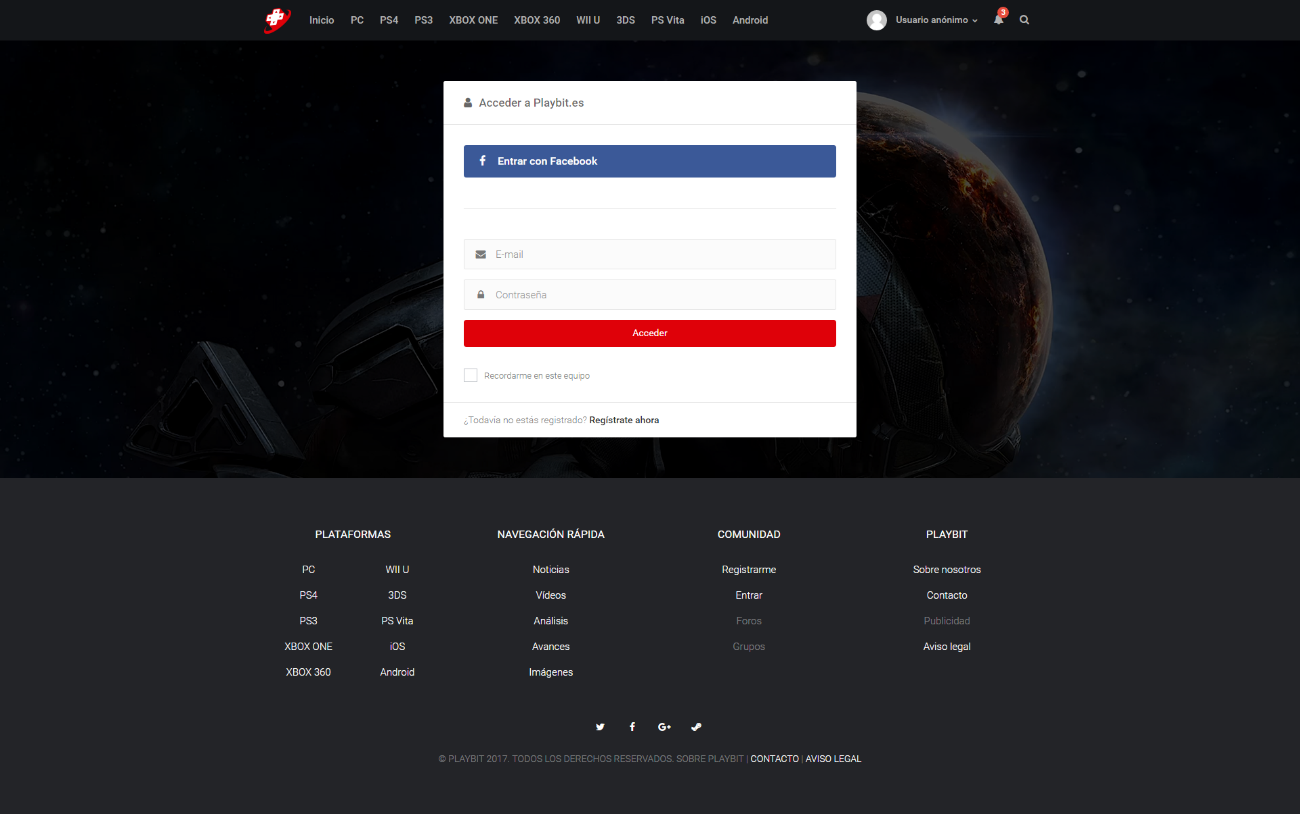
**Registro de usuarios**

Descripción



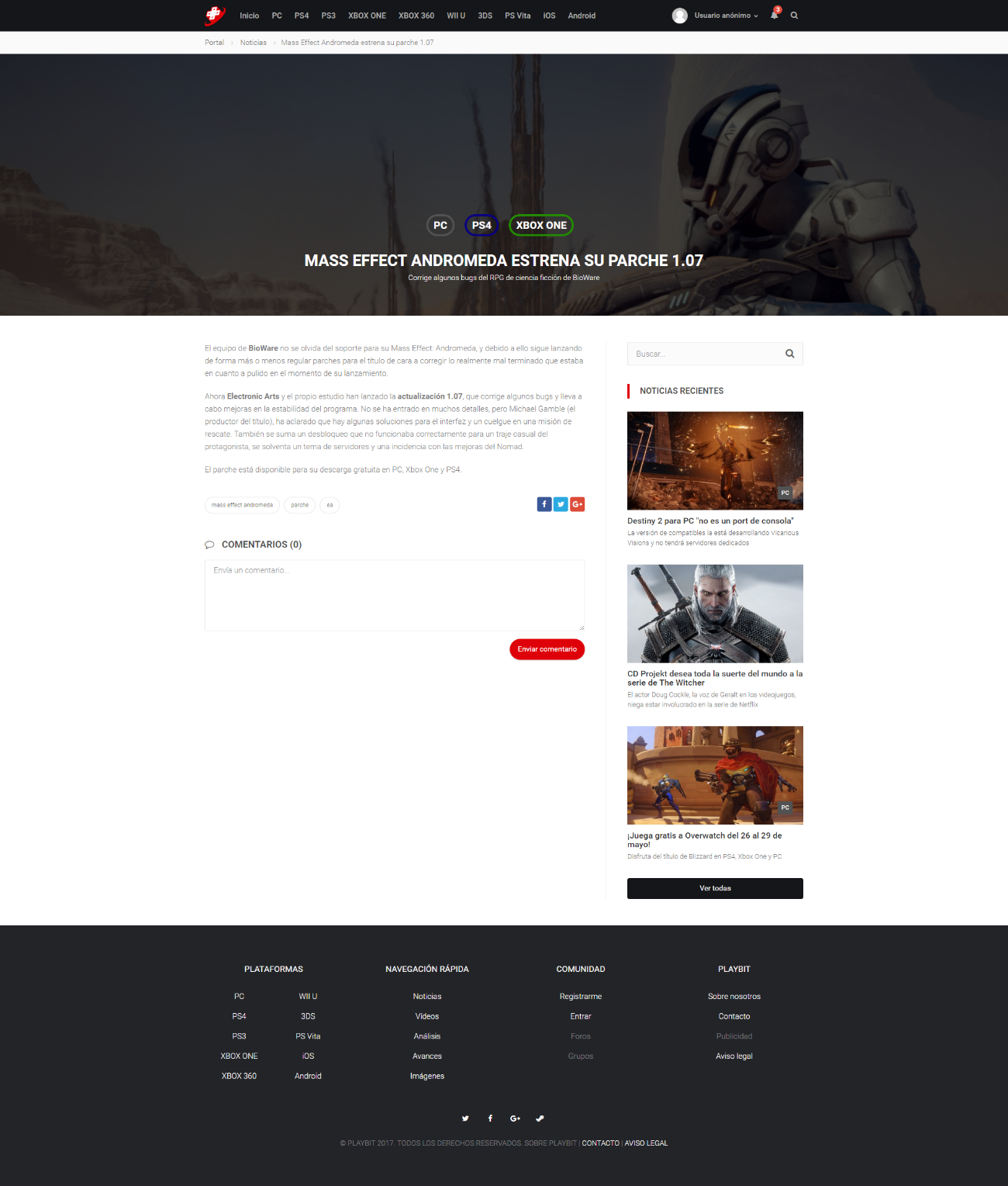
**Acceso de usuarios registrados**

Descripción



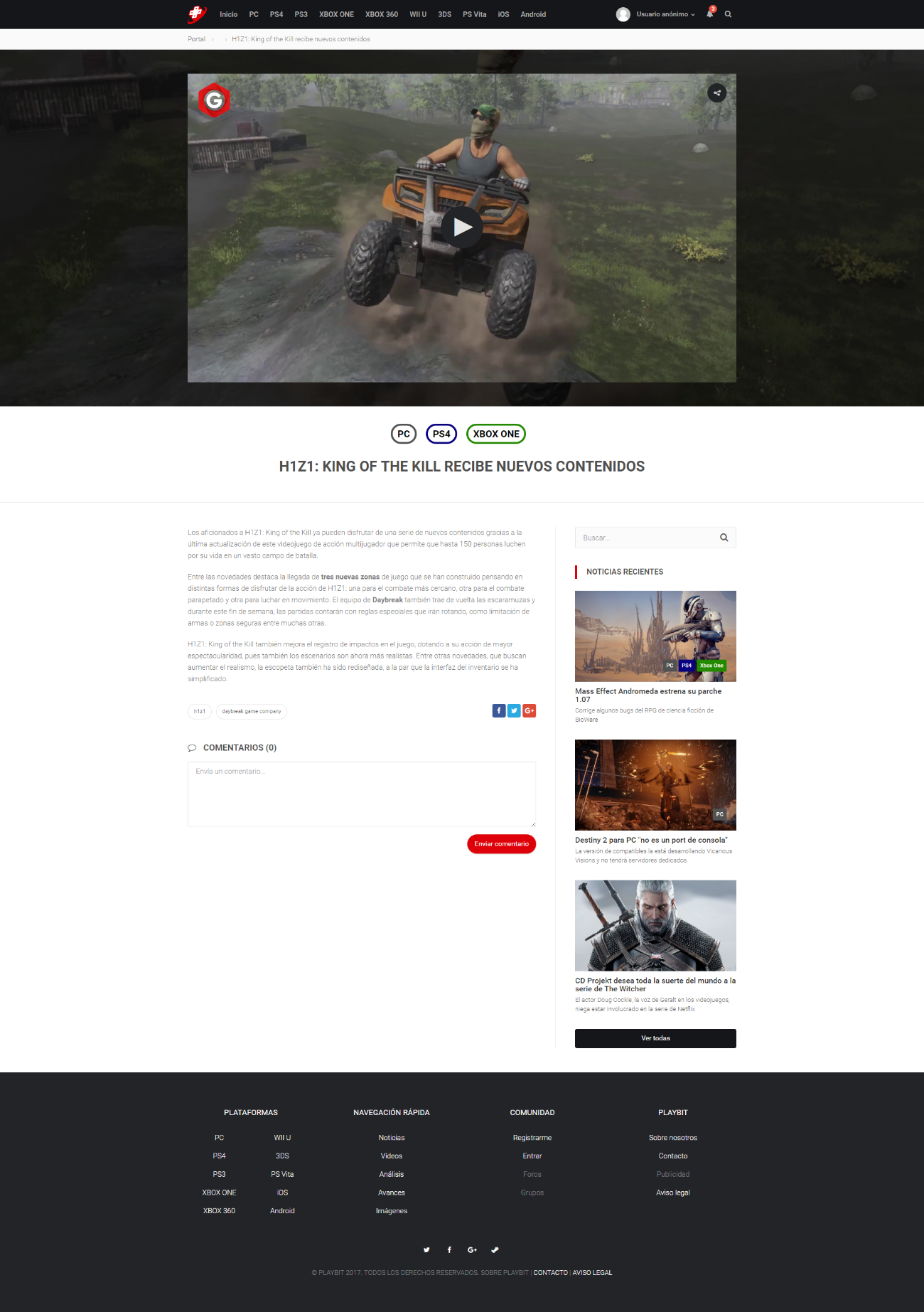
**Noticia**

Descripción

****

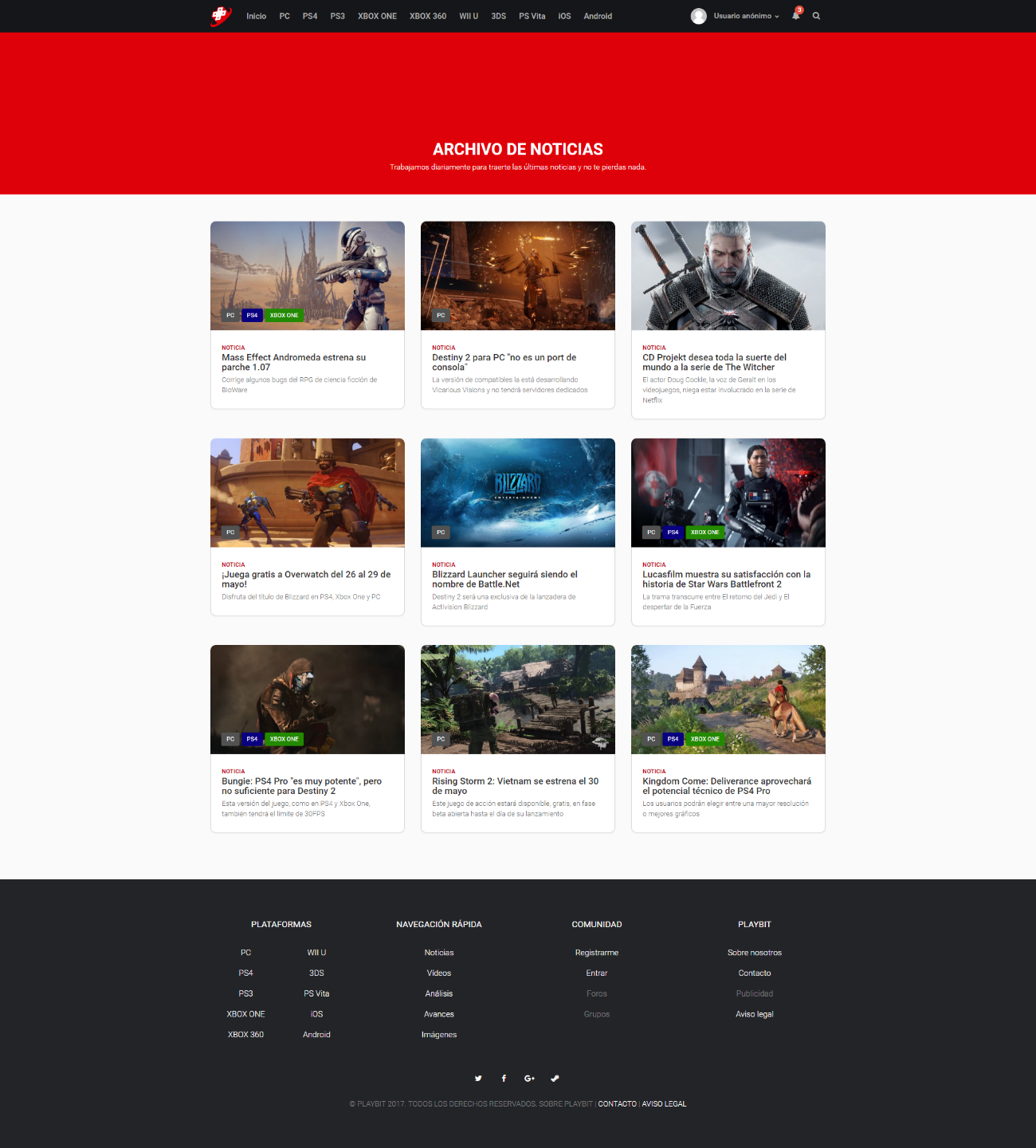
**Vídeo**

Descripción



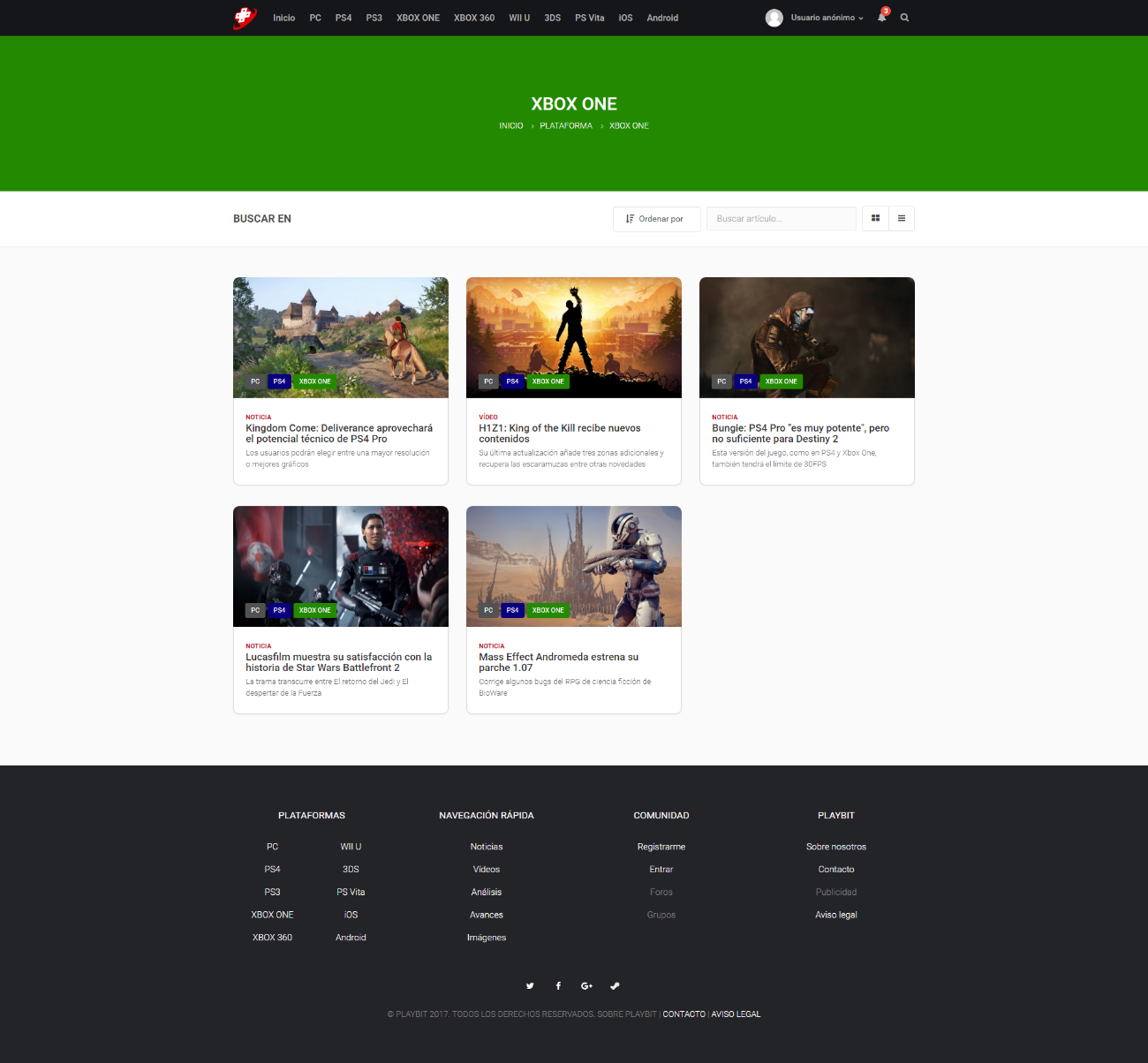
**Archivo de noticias**

Descripción



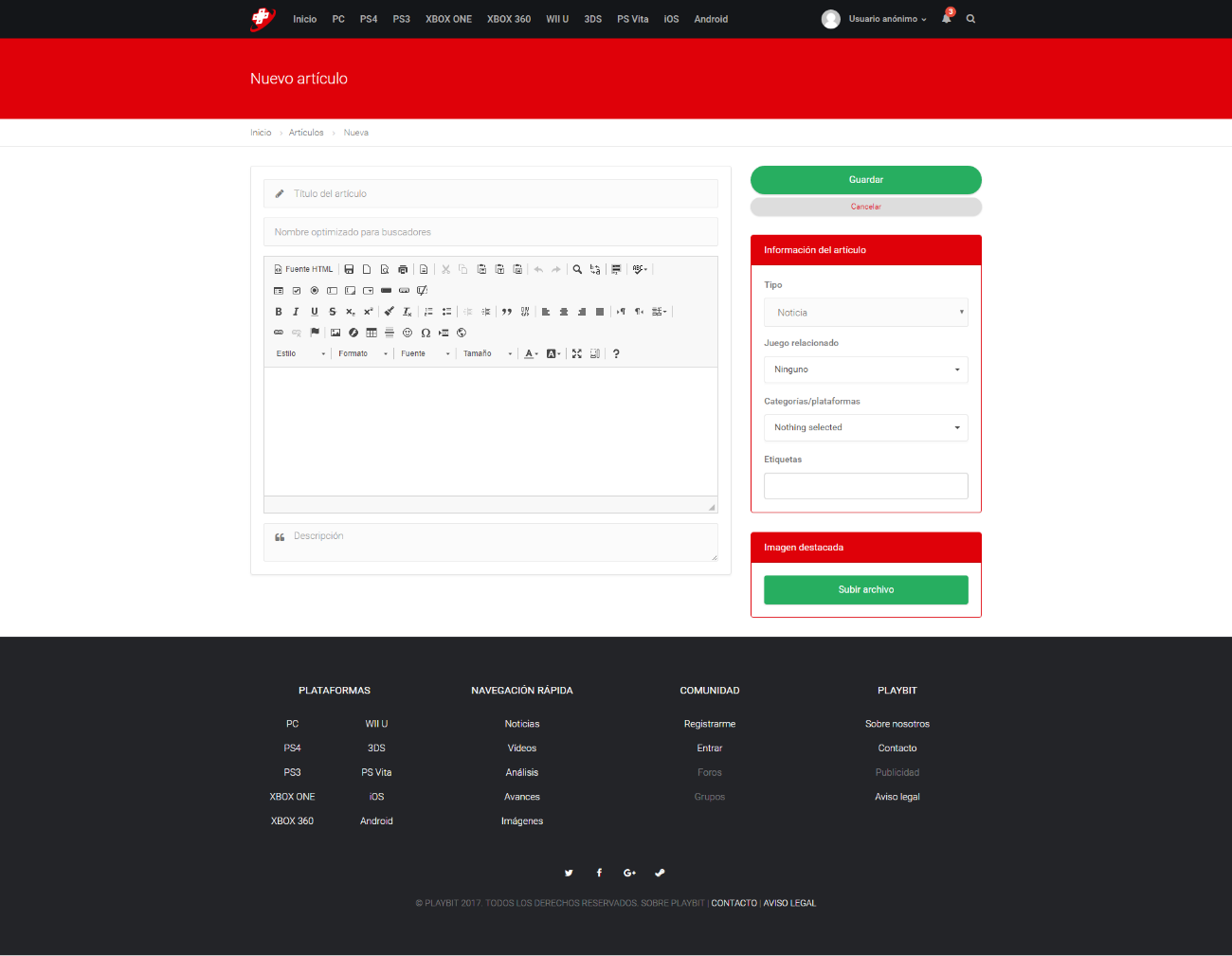
**Categoría/Plataforma**

Descripción



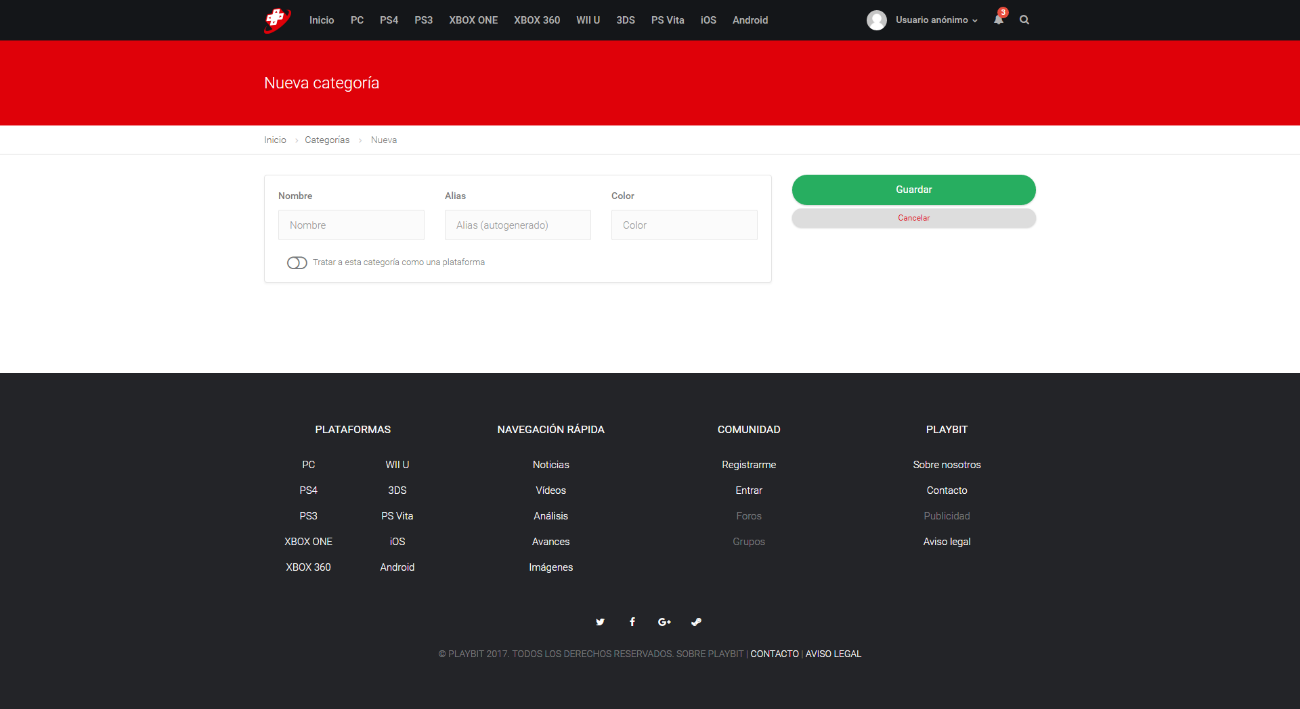
**Administración – Crear un nuevo artículo**

Descripción



**Administración – Crear una nueva categoría**

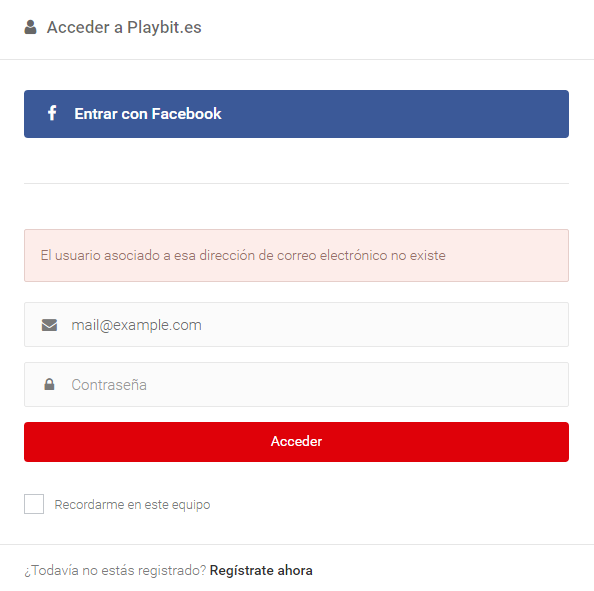
Descripción



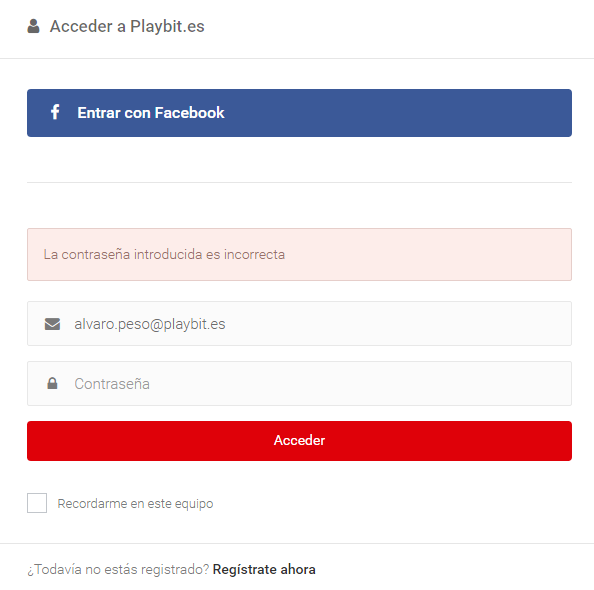
## Pruebas

**Acceso de usuarios**

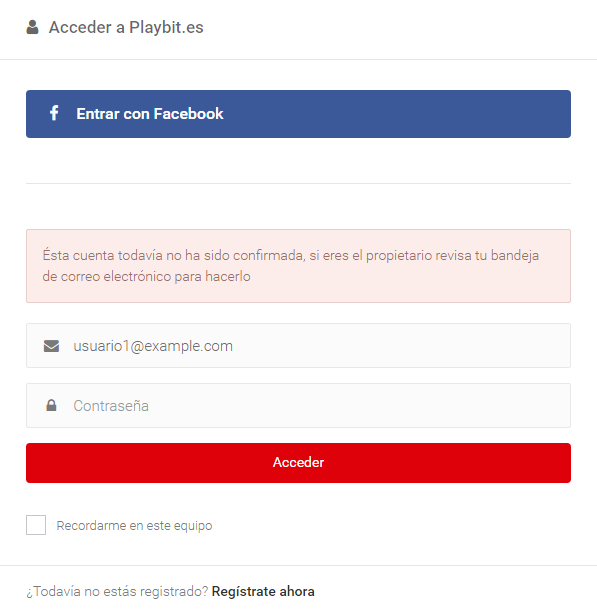
Si un usuario intenta acceder con un correo electrónico que no está asociado a ningún usuario.



Si la contraseña introducida es incorrecta.



Si el usuario intenta acceder con una cuenta no verificada.



**Confirmación de correo electrónico**

Cuando un usuario se registre, deberá verificar su e-mail mediante un enlace que recibirá en un correo electrónico como el siguiente.

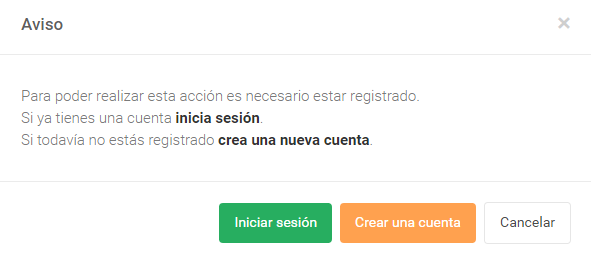


****

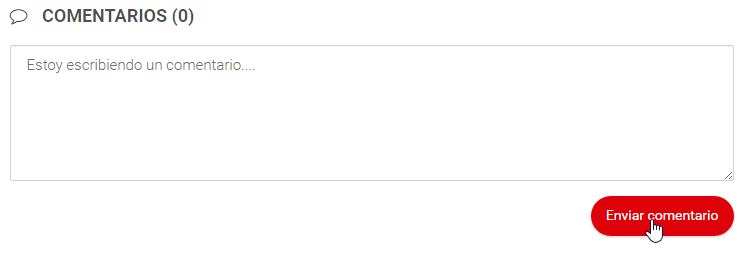
Una vez confirmado, este usuario ya podrá acceder a la aplicación con su correo electrónico y contraseña.

**Añadir nuevos comentarios en los artículos**

Si un usuario intenta escribir un comentario en un artículo sin estar autenticado, se mostrará una alerta invitándole a acceder al sistema o si todavía no lo ha hecho, a registrarse.

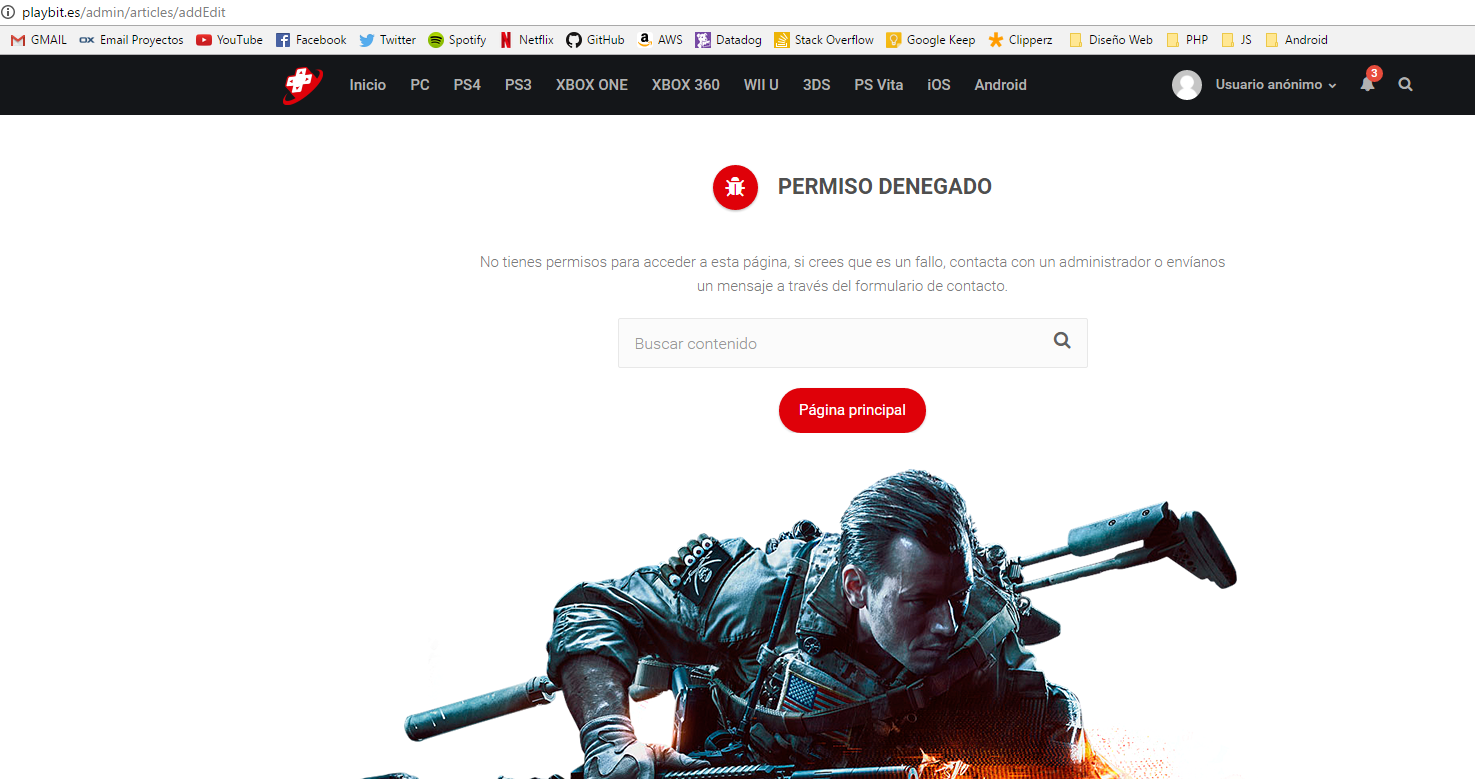
****

Si el usuario está autenticado podrá comentar en artículos.

****

**Roles y permisos**

Si un usuario no tiene permisos para acceder a una página determinada, se lanzará una excepción y se mostrará una pantalla de error.

****

# Conclusiones finales

## Grado de cumplimiento de los requisitos fijados y análisis de las desviaciones

## Propuestas de mejora o ampliaciones futuras

# Guías

## Guía del usuario

## Guía del administrador

# Referencias bibliográficas