



Escuela  
Politécnica  
Superior

# Plataforma de valoración y venta de consolas, juegos y accesorios: versión WEB



Grado en Ingeniería Informática

## Trabajo Fin de Grado

Autor:

Rafael José González Palomares

Tutor/es:

Jaume Aragones Ferrero



Universitat d'Alacant  
Universidad de Alicante

Septiembre 2016

# ÍNDICE

<b>INTRODUCCIÓN .....</b>	<b>4</b>
<b>Descripción del proyecto.....</b>	<b>4</b>
<b>Marco Teórico.....</b>	<b>5</b>
<b>OBJETIVO .....</b>	<b>6</b>
<b>TECNOLOGÍAS Y HERRAMIENTAS EMPLEADAS.....</b>	<b>6</b>
<b>Trello.....</b>	<b>6</b>
<b>phpMyAdmin.....</b>	<b>7</b>
<b>Visio .....</b>	<b>7</b>
<b>Lucidchart.....</b>	<b>7</b>
<b>PhpStorm .....</b>	<b>7</b>
<b>Mockups .....</b>	<b>8</b>
<b>METODOLOGÍA .....</b>	<b>8</b>
<b>SISTEMAS DE INFORMACION.....</b>	<b>10</b>
<b>Base de Datos .....</b>	<b>10</b>
<b>Diagrama de Clase.....</b>	<b>12</b>
<b>Casos de uso .....</b>	<b>13</b>
Público .....	14
Privado.....	16
Administrador.....	18
<b>Mockups .....</b>	<b>20</b>
Publica .....	21
Privado.....	26
Administrador.....	28
<b>IMPLEMENTACIÓN .....</b>	<b>34</b>
<b>Implementación API .....</b>	<b>34</b>
<b>Implementación Web .....</b>	<b>37</b>
Modelo Vista Controlador .....	37
Empezando Symfony.....	38
Controladores.....	39
Vistas .....	40
Modelo.....	41

Formularios.....	42
<b>Implementación imágenes .....</b>	<b>44</b>
Rol Público .....	44
Rol Privado .....	46
Rol Administrador .....	46
<b>SEGURIDAD.....</b>	<b>47</b>
<b>Problemas principales en la programación de sistemas web .....</b>	<b>47</b>
<b>Prácticas básicas de seguridad web .....</b>	<b>47</b>
Balancear riesgo y usabilidad .....	47
Rastrear el paso de los datos .....	48
Filtrar entradas .....	48
Escapado de salidas .....	48
<b>Métodos empleados para mejorar la seguridad .....</b>	<b>48</b>
Contraseña del usuario.....	48
Ataques de inyección SQL .....	49
Protección <i>CSRF</i> .....	49
<b>PRUEBAS .....</b>	<b>49</b>
<b>PROBLEMAS Y DIFICULTADES .....</b>	<b>50</b>
<b>Dificultad.....</b>	<b>50</b>
<b>Problemas.....</b>	<b>50</b>
<b>CONCLUSIONES.....</b>	<b>51</b>
<b>Mejoras y ampliaciones.....</b>	<b>52</b>
<b>BIBLIOGRAFÍAS Y REFERENCIAS .....</b>	<b>53</b>
<b>ANEXO .....</b>	<b>54</b>
<b>Anexo 1 .....</b>	<b>54</b>
Diccionario de datos .....	54
<b>Anexo 2 .....</b>	<b>59</b>
Modelo Lógico.....	59
<b>Anexo 3 .....</b>	<b>64</b>
Procedimientos .....	64

# INTRODUCCIÓN

## *Descripción del proyecto*

La idea del proyecto es crear un plataforma que sirva para tasar productos dedicados al mundo de los videojuegos. Dando a los usuarios la facilidad de obtener valoraciones sobre el precio de los productos, y permitiendo que puedan efectuarse transacciones de compra y venta desde la misma plataforma. Facilitando así ambas tareas desde la misma web.

El eje principal del proyecto busca analizar, diseñar e implementar un aplicación web que permita valorar, comprar y vender consolas, accesorios y videojuegos.

Además los usuarios podrán tasar su producto una vez iniciada la sesión y podrán publicarlo en la web totalmente gratis.

Para el desarrollo de la aplicación web se ha utilizado como herramienta principal el framework **Symfony** que explicamos a continuación.

Symfony es un completo framework diseñado para optimizar el desarrollo de las aplicaciones web basado en el patrón Modelo Vista Controlador (MVC), que funciona a partir de un patrón de arquitectura de software. Además de separar los datos y la lógica de negocio de la interfaz de usuarios y a su vez del módulo de comunicaciones y eventos.

Este tipo framework proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. Lo que ha facilitado la tarea a lo largo del desarrollo de la web.

Symfony está desarrollado completamente en PHP 5.3. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Es compatible con la mayoría de gestores de bases de datos y se puede ejecutar tanto en plataformas (Unix, Linux, etc.) como en plataformas Windows.



## *Marco Teórico*

Lo que se intenta conseguir con esta web es suplir la ausencia de tasadores online, ya que el hecho de que la tecnología es un campo que varía diariamente. Por ello todos los productos tecnológicos cambian su valor en un corto periodo de tiempo, algunos se revalorizan y otros en cambio pierden valor.

Por ello el objetivo central de esta web es proporcionar un servicio de tasación online, en este caso de videojuegos, consolas y accesorios donde los usuarios además de poder tasar su producto podrán comprar y vender otros productos.

En el mercado actualmente no hay nada parecido a GAME2SELL, se ha encontrado un tasador online similar pero que tasa y ofrece otro tipo de productos, por lo que esta web sería única en el campo de los videojuegos ofreciendo este servicio.

A lo largo de la búsqueda bibliográfica previa a la realización del proyecto encontramos otras webs similares a lo que busca parecerse GAME2SELL.

En la web Mac2Sell encontramos un servicio online de tasación de dispositivos Apple. Esta aplicación permite a los usuarios conocer los precios de ordenadores usados de esa misma casa. Un servicio de mucha utilidad para cualquier interesado en comprar o vender un ordenador de esta marca.

A nivel usuario podemos seleccionar las características de nuestro interés y saber cuál es el precio de dicho dispositivo. Una vez seleccionadas las características la página tasa el valor del dispositivo basándose en el valor que marca el mercado.

## OBJETIVO

El objetivo principal del proyecto es crear una plataforma que permita tasar todo tipo de productos relacionados con el campo de los videojuegos. Buscamos analizar, diseñar e implementar una web que ofrezca a los usuarios la opción de valorar, comprar y vender cualquier producto relacionado con los videojuegos.

Se trata de desarrollar una aplicación web compatible con los navegadores de internet, consiguiendo así que cualquier usuario pueda hacer uso de la misma, y que a su vez nuestra web se convierta en un referente a la hora de realizar una compra/venta de artículos relacionados con los videojuegos.

El sistema de tasación se ha basado en una media establecida a partir de otras webs del mercado que ofrecen nuestros productos. Pero a diferencia de dichas webs GAME2SELL busca la interacción con los usuarios, ofreciendo así la oportunidad de que ellos mismos propongan los precios de cada producto. A pesar de que esta característica nos diferencia de otras webs de tasación cabe destacar la importancia de que el tasador debe ser capaz de adaptar el valor de los artículos al ritmo que marca el mercado.

## TECNOLOGÍAS Y HERRAMIENTAS EMPLEADAS

### *Trello*



Trello es una herramienta que se ha utilizado a modo de panel Kanban, creando tarjetas asociadas a tareas para darle visibilidad al desarrollo del software, es decir, saber cuánto se ha hecho, y cuanto falta por realizar. Consiguiendo así un mayor control sobre el proceso de desarrollo software y cumplir con el objetivo de entregar el proyecto a tiempo en el plazo acordado.

## PhpMyAdmin

PhpMyAdmin es la herramienta que se ha utilizado para el manejo de la base de datos MySQL está alojado en el servidor junto con el Api y la página web. La versión de MySQL es la 5.5.50 y la versión de phpMyAdmin utilizada es la 4.2.12 y está alojada sobre un servidor Apache versión 2.2.31.



## Visio

Visio es una herramienta para la realización de diagramas que nos permite simplificar y comunicar información compleja. En nuestro caso la utilizamos para realizar el lenguaje unificado de modelado (UML), donde especificamos y describimos los métodos y procesos utilizados en la parte de la base de datos.



## Lucidchart

Para la realización de los casos de usos se ha utilizado la herramienta Lucidchart, dicha herramienta nos permite realizar los diferentes casos de uso de manera colaborativa, en la cual muchos participantes pueden ver, editar y borrar al mismo tiempo.



## PhpStorm

**PhpStorm** es un IDE de programación de entornos más completos de la actualidad, permite editar código no sólo del lenguaje de programación PHP. Permite la gestión de proyectos fácilmente, proporciona un fácil **autocompletado de código**.



En PhpStorm la **ejecución de nuestro código** es en la misma interfaz del IDE. Así como también la interpretación y **visualización inmediata** de código.

Útil para para los **Frameworks** como Laravel o Symfony los cuales trabajan a menudo con "**terminales**" para *crear, implementar, eliminar*, controladores, modelos, etc.

## Mockups

Para la realización de los mockups se ha utilizado la herramienta online **mocqups**, en la cual se han diseñado los mockups separando por roles. Los beneficios de esta herramienta es que sirve para todo tipo de mockups y al ser online se puede trabajar de manera colaborativa



## METODOLOGÍA

Antes de explicar la metodología utilizada para la realización de la aplicación web cabe mencionar que, además la aplicación web GAME2SELL también se han realizado por parte de otros compañeros la aplicación para IOS y Android.

En las 3 variedades de plataforma se ha partido de una base de datos común buscando así que los usuarios puedan acceder desde cualquier dispositivo sin que varíe el servicio que se ofrece desde GAME2SELL, favoreciendo a los usuarios registrados el acceso desde dispositivos móviles.

Como se ha comentado anteriormente la base de datos común entre los 3 trabajos ha sido realizada por 3 personas en conjunto. Para su realización se decidió utilizar la metodología SCRUM. Posteriormente se establecieron las tareas a realizar por cada parte (backlog).

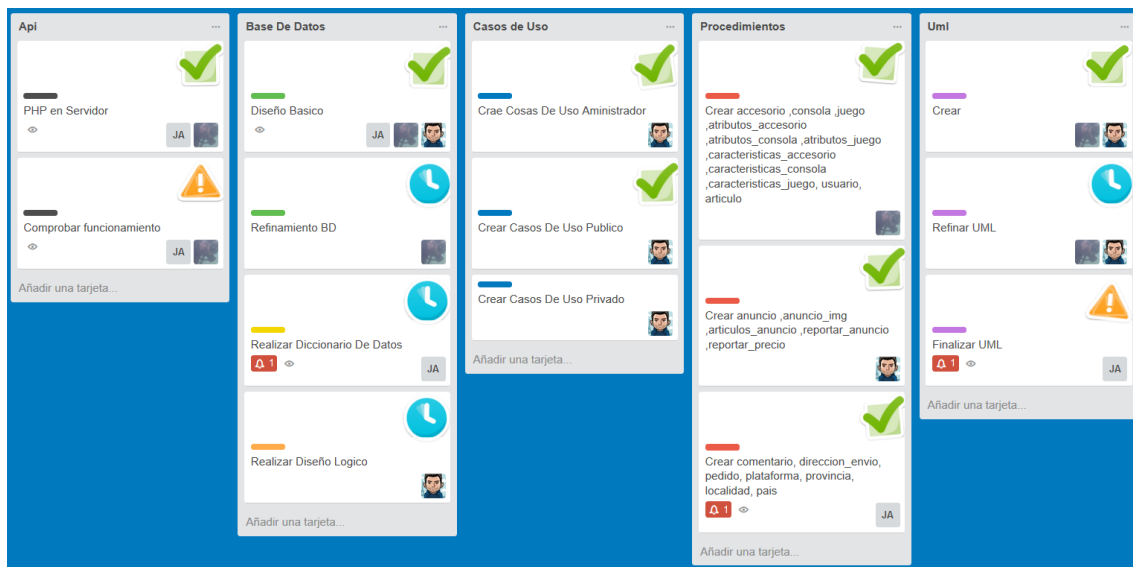
La comunicación entre todas las partes ha sido la clave para que se realizara una base de datos con éxito. Durante todo el proceso se han ido realizando varios sprints por parte de cada uno y durante estos sprints hemos ido realizando dailys sprints durante los que se planteaban los problemas que habían ido surgiendo y se planificaban las tareas siguientes a realizar por cada integrante.

Con estas reuniones lo que se ha buscado es comprobar que se estaban realizando todas las tareas pertinentes y que todo los puntos iban saliendo con éxito, además de que la comunicación ha favorecido a la resolución de dudas, inseguridades y errores.

Para poder llevar al día el estado real de la parte común del proyecto, en este caso la base de datos, se creó un panel kanban, donde se ha podido observar de una manera sencilla las tareas que hemos ido realizando, las que aún no estaban completadas y las que quedaban por completar. Ofreciéndonos una visión global del estado de las tareas y una visualización del recorrido hasta conseguir los objetivos.

Se muestra a continuación un ejemplo de cómo ha funcionado el panel:

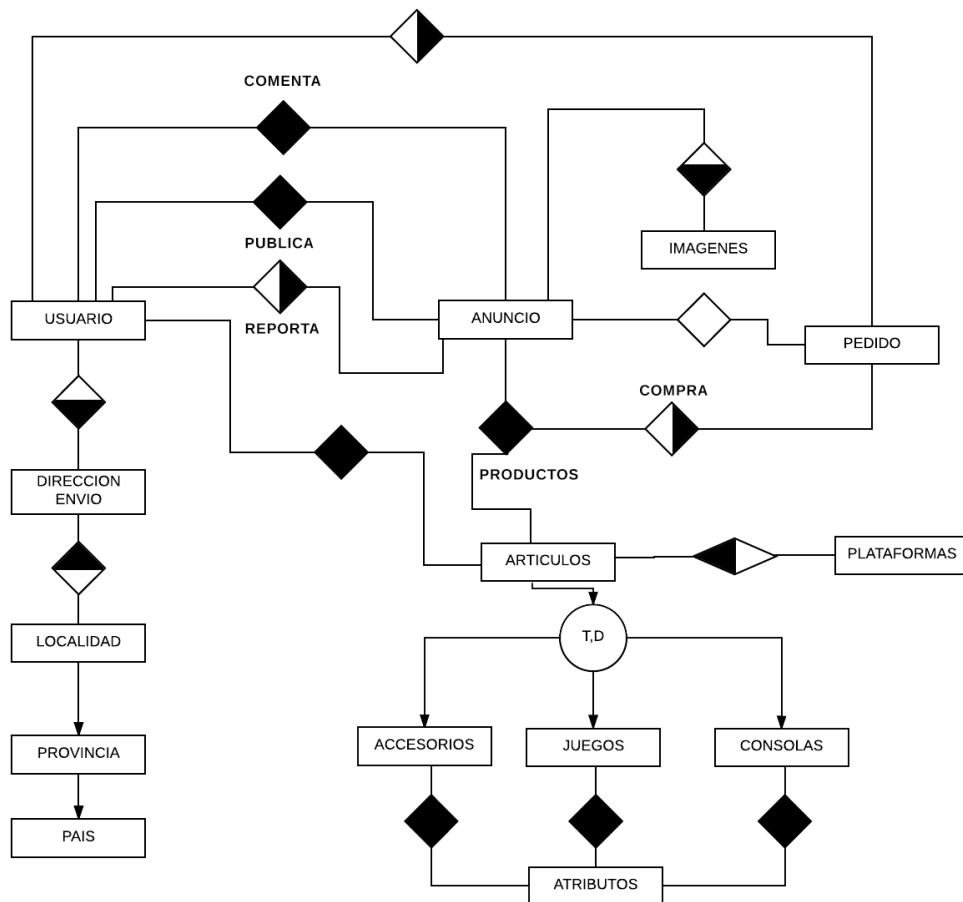
- Check verde :Las tareas marcadas que están finalizadas.
- Reloj azul: Las tareas que se encuentran en proceso
- Advertencia: Las tareas que aún no han sido comenzadas.



# SISTEMAS DE INFORMACIÓN

## Base de Datos

La base de datos utilizada en GAME2SELL se ha realizado mediante el modelo relacional basado en los estándares de las asignaturas de Fundamentos de base datos y diseño de base de datos. Siendo este el más conocido por el grupo.



Destacar del modelo relacional la generalización de los atributos en la entidad artículos, estos a su vez tendrán unos determinados atributos:

- Consola: tipo, modelo, capacidad, formato y color.
- Juegos: formato, idioma, genero, desarrollador y edad.
- Accesorios: capacidad y color

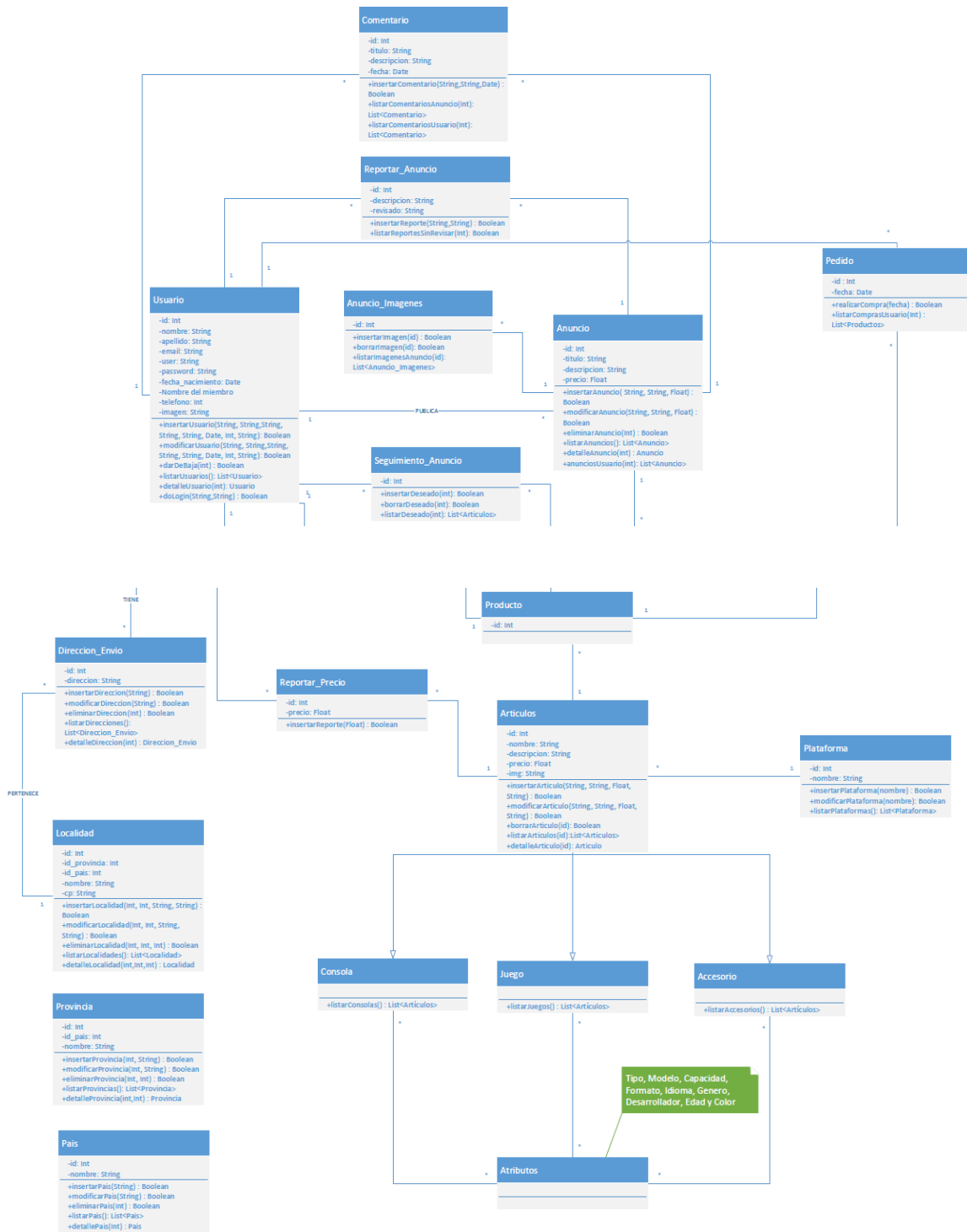
También aclarar la parte de país, provincia y localidad, ya que se ha preparado la base de datos para la extensión a otros países y para que el tasador tenga más variedad a la hora de obtener la tasación de un producto determinado.

Partiendo del modelo relacional se ha desarrollado el diccionario de datos que se encuentra en el ANEXO 1 y el modelo lógico de la base de datos que se puede ver en el ANEXO 2.

Una vez finalizada la base de datos ya puede ser implementada en un servidor físico. Para ello ha sido desarrollada con el lenguaje MYSQL y utilizado PHPMYADMIN para manejar y administrar la base de datos. Además para facilitar la tarea de los desarrolladores, evitando continuas consultas al servidor, se han creado unos procedimientos que se encuentran en el ANEXO 3.

## Diagrama de Clase

Con el objetivo de facilitar la comunicación y obtener una representación de forma gráfica del problema y su solución, se ha creado un diagrama de clases donde especificamos con una estructura precisa las decisiones de análisis, diseño e implementación del sistema de datos.





## Casos de uso

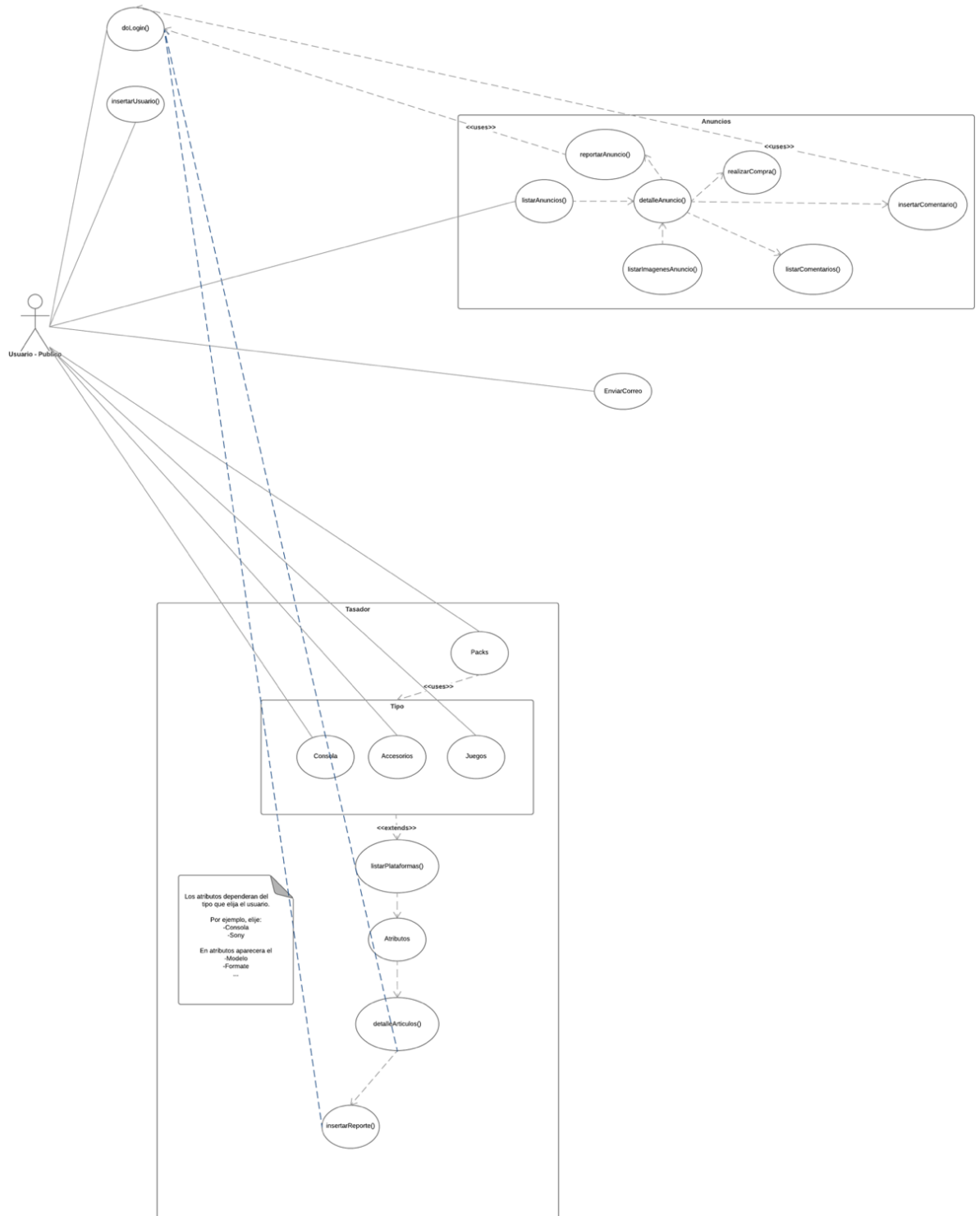
Se han diseñado varios diagramas que muestran la relación entre los actores y los casos de uso en la aplicación web separando los roles. Encontramos 3 tipos de roles diferentes:

- **Público:** Hace referencia al rol que ejerce un usuario sin haber iniciado sesión. Dentro de este rol tienes acceso a diferentes servicios dentro de la aplicación como acceder a la información que ofrecen los anuncios y la información de las tasaciones.
- **Privado:** Hace referencia al rol que ejerce un usuario una vez que ha iniciado sesión. Dentro de este rol puedes acceder a tu cuenta, donde podrás modificar información acerca de tu perfil, también podrás acceder a los anuncios y una vez conocida la tasación podrás comprar y vender.
- **Administrador:** Desde este rol el administrador tiene acceso a los artículos, a los anuncios, a los usuarios, también a los reportes y los pedidos. De manera que desde su inicio puede gestionar y modificar toda la información pertinente a estas áreas.

A continuación se expone una muestra de las funciones que tiene cada rol, explicándolas brevemente.

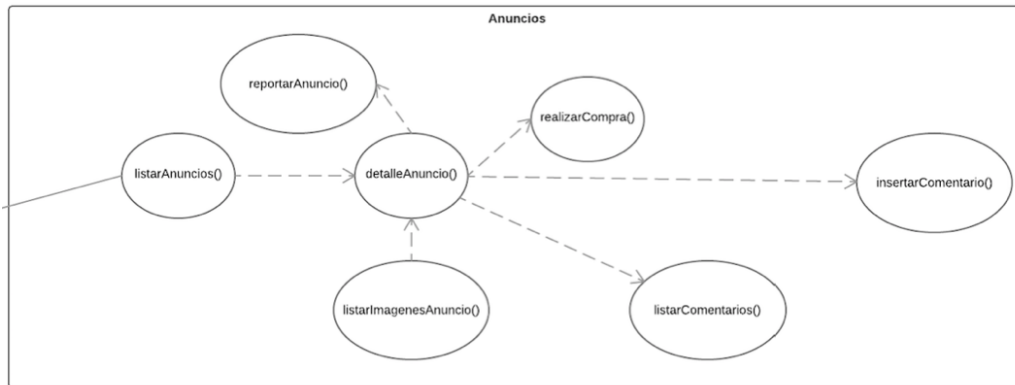
## Público

Este rol hace referencia a los usuarios que no han iniciado sesión en la web.



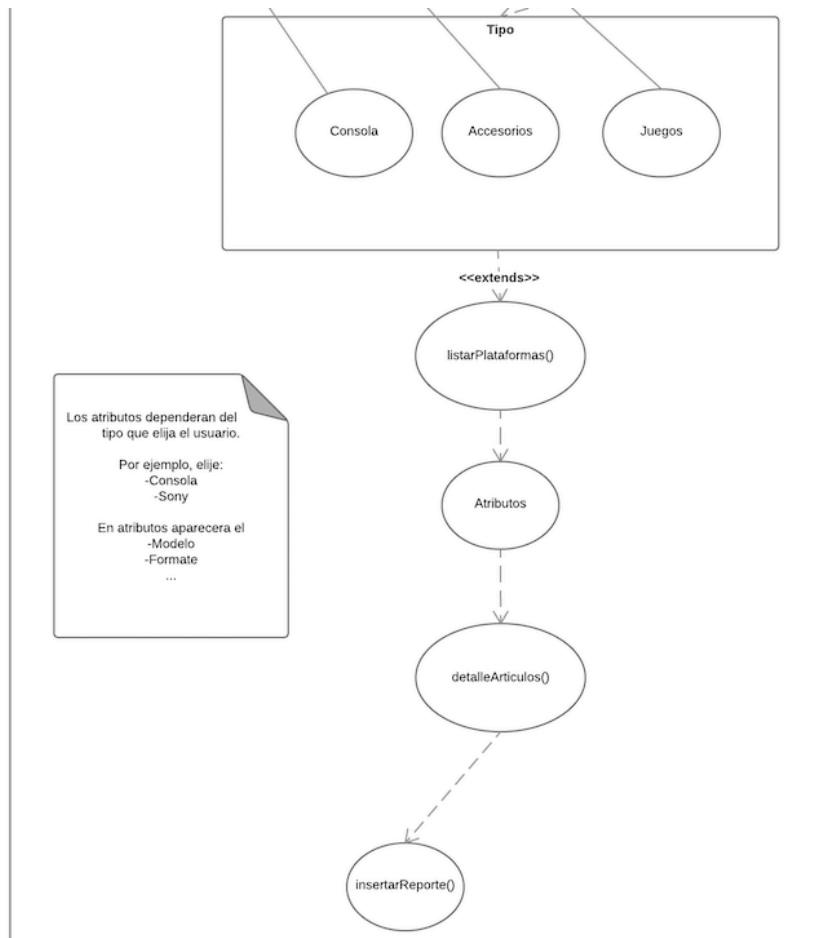
▪ Anuncios

En esta parte del diagrama se pueden observar todas las acciones a las que tiene acceso el usuario sin haber iniciado sesión, como obtener información de los anuncios de la web, características del producto anunciado, etc.



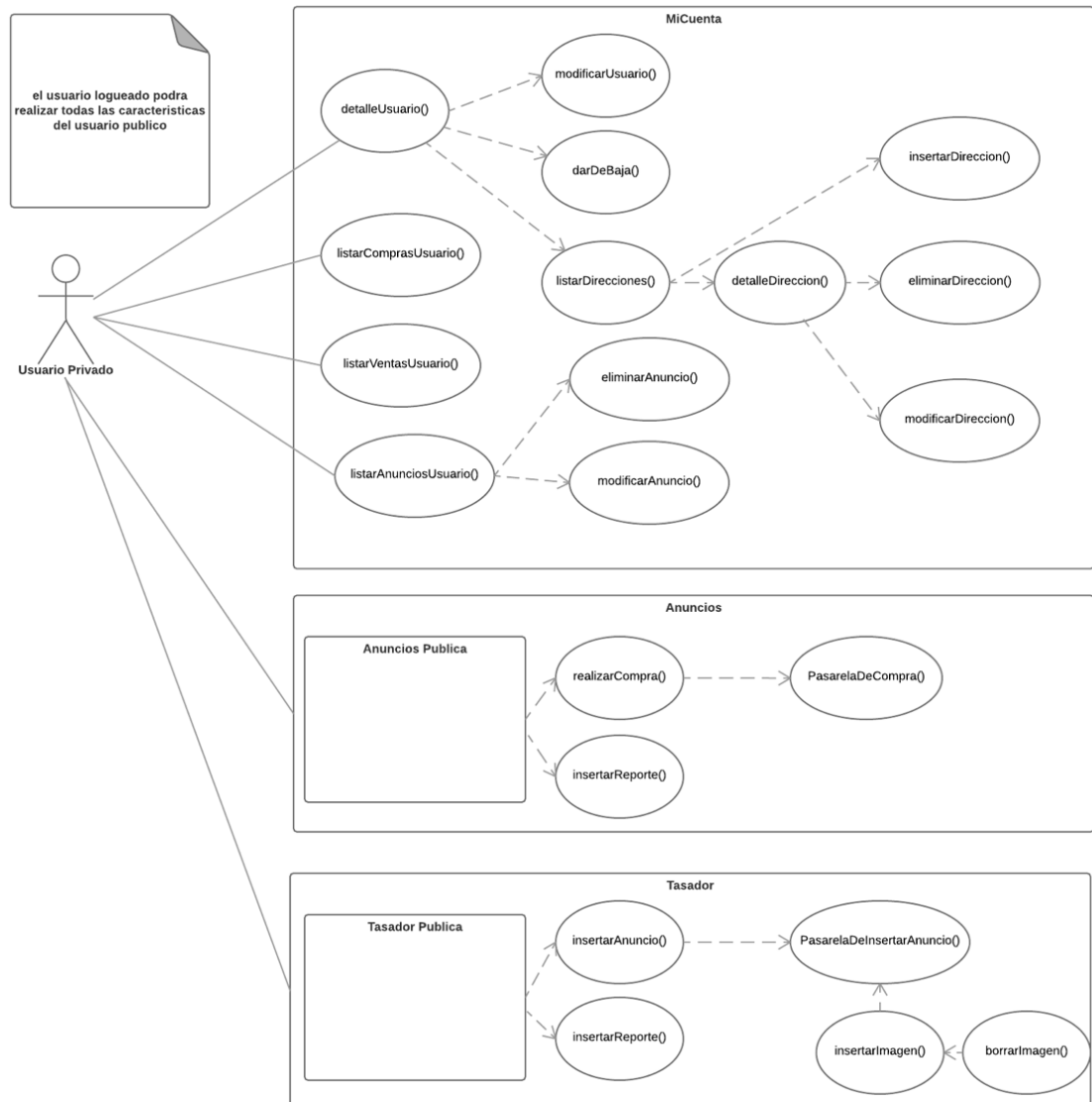
▪ Tasador

El usuario podrá consultar las tasaciones de los productos sin necesidad de haber iniciado sesión, además de consultar el precio también podrá obtener información del artículo tasado.



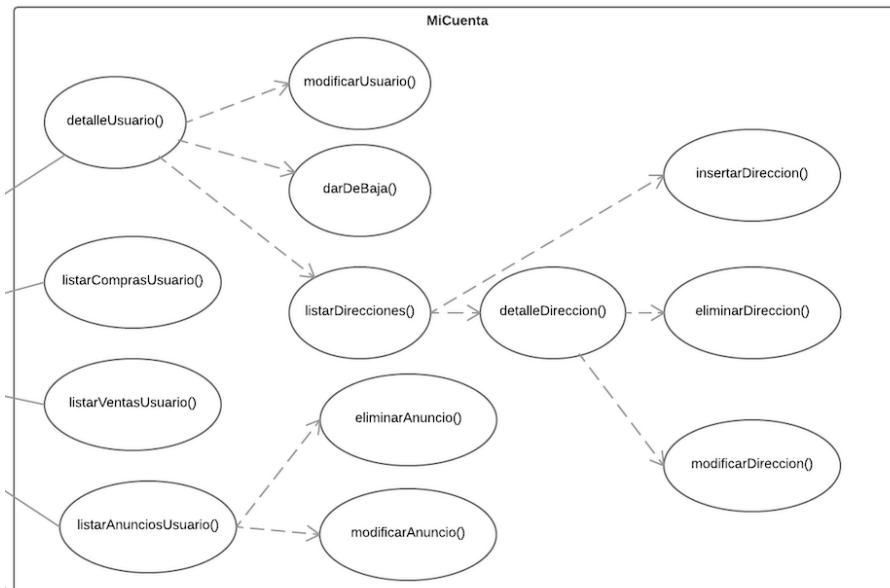
## Privado

Este rol está compuesto por los usuarios que inician sesión dentro de la web. En este apartado encontramos diferentes acciones que el usuario puede realizar una vez registrado.



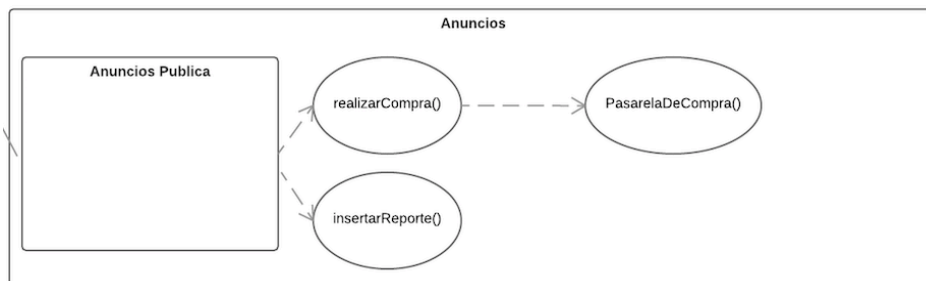
### ▪ Mi Cuenta

En este apartado el usuario que ha iniciado sesión puede acceder a sus datos personales, pudiendo hacer modificaciones como darse de baja de la aplicación, ver las direcciones de los envíos, así como insertar, eliminar y modificarlas



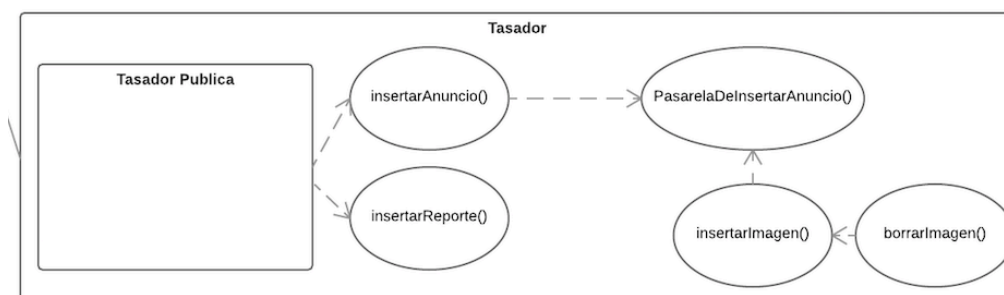
- Anuncios

Al igual que en la parte pública, el usuario registrado puede acceder a los mismos anuncios pero además de obtener información de los productos puede realizar el feedback al administrador si algún anuncio no está correctamente. Una vez obtenida la información del producto puede realizar la compra del mismo.



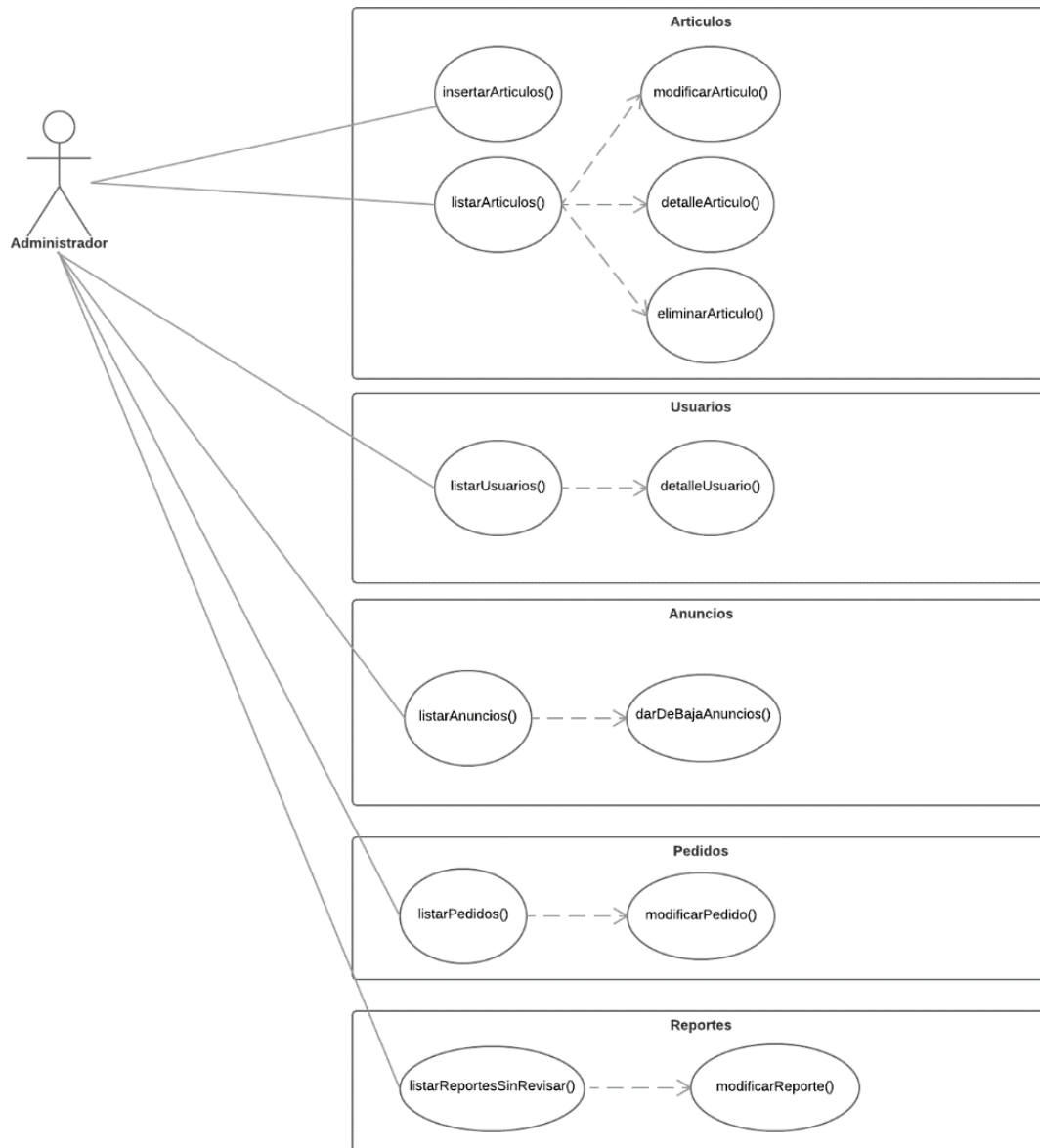
- Tasador

Al igual que en la parte pública el usuario registrado puede publicar un anuncio y realizar el reporte del precio como rol de tasador.



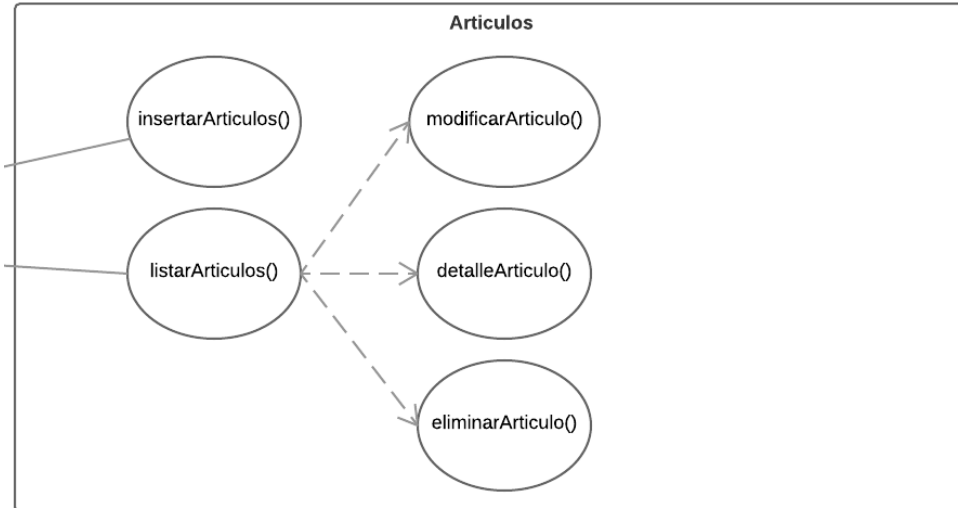
## Administrador

En este rol encontramos diferentes apartados que se han dividido según las acciones que puede realizar el administrador: Insertar artículos, dar de baja usuarios, eliminar anuncios, gestionar pedidos y comprobar los reportes de los usuarios. Mediante estas acciones se consigue que la web este en perfecto funcionamiento.



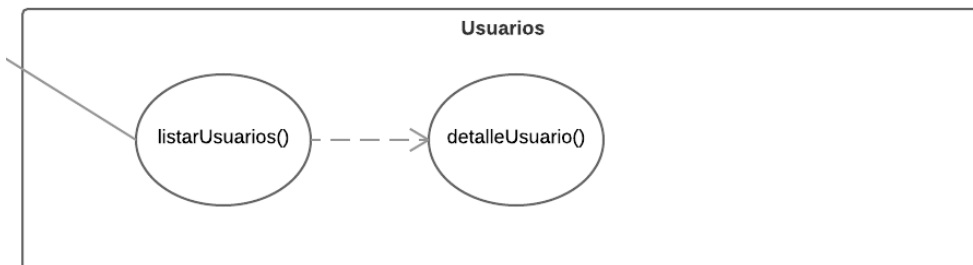
- Artículos

El administrador listará los artículos que se encuentran en la base de datos, desde la cual podrá obtener la información necesaria para que el tasador funcione correctamente.



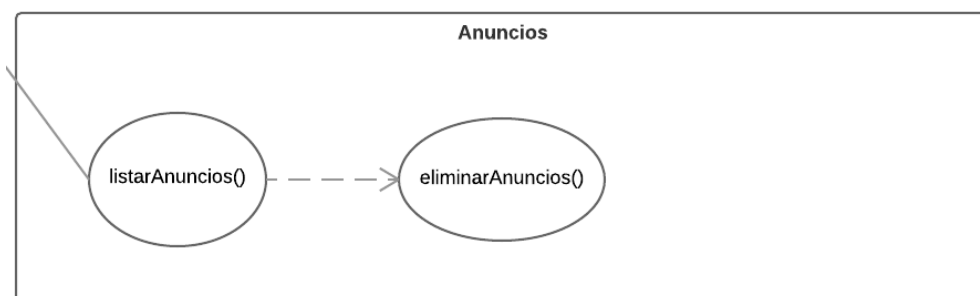
- Usuarios

El administrador obtiene una lista de todos los usuarios registrados y puede ver los detalles de cada usuarios comprobando que no haya incidentes.



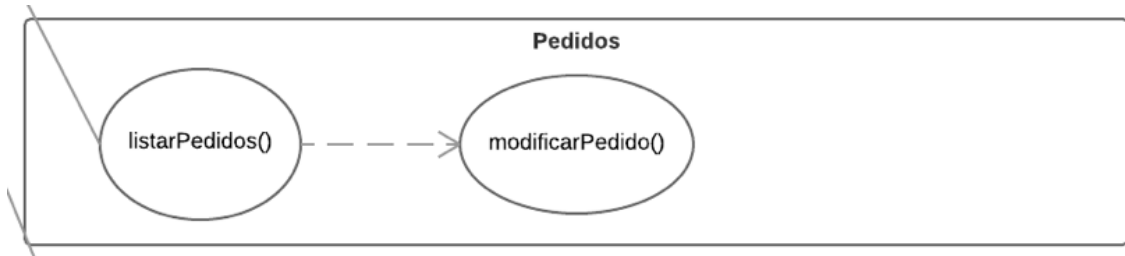
- Anuncios

El administrador es el encargado de gestionar los anuncios de la página, pudiendo modificar lo que se necesario.



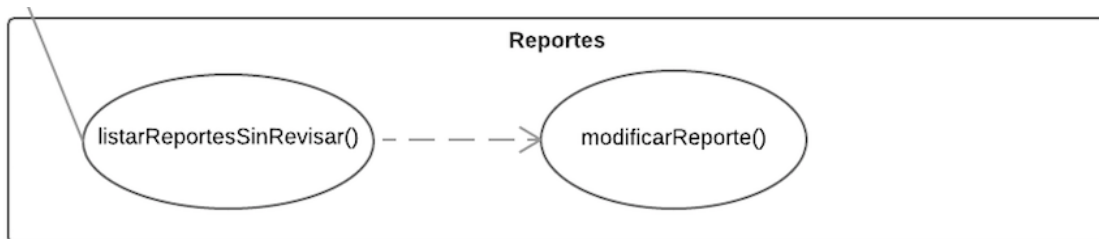
- Pedidos

El administrador podrá ver todos los pedidos que se han realizado y podrá modificar su estado para que el usuario compruebe en qué estado está el pedido realizado.



- Reportes

El administrador es el encargado de gestionar los reportes realizados por los usuarios. Ayudando esto a la interacción entre la administración y gestión de la página y sus usuarios.



## Mockups

Se han diseñado los mockups agrupados en función de los 3 roles que parecen en nuestra aplicación y coincidiendo con los roles de los casos de uso.

- Rol público, accesible para cualquier persona que entre en la página web.
- Rol privado, para un usuario que haya iniciado su sesión. Donde es posible acceder a sus datos personales.
- Rol administrador, encargado de mantener este sitio y cuyas características explicaremos más adelante.



- Cabecera

Durante el uso de la página web el usuario podrá encontrar el menú en la cabecera, ésta aparecerá la parte superior de la pantalla en todo momento, facilitando así la entrada en las diferentes partes de aplicación. En la imagen se pueden observar las diferentes opciones del menú.



Si el usuario inicia sesión el menú será el mismo, pero además tendrá la opción de acceder a su perfil. Y aparecerá su nombre de usuario y su identificación en la esquina superior izquierda, como se puede ver a continuación.



## Publica

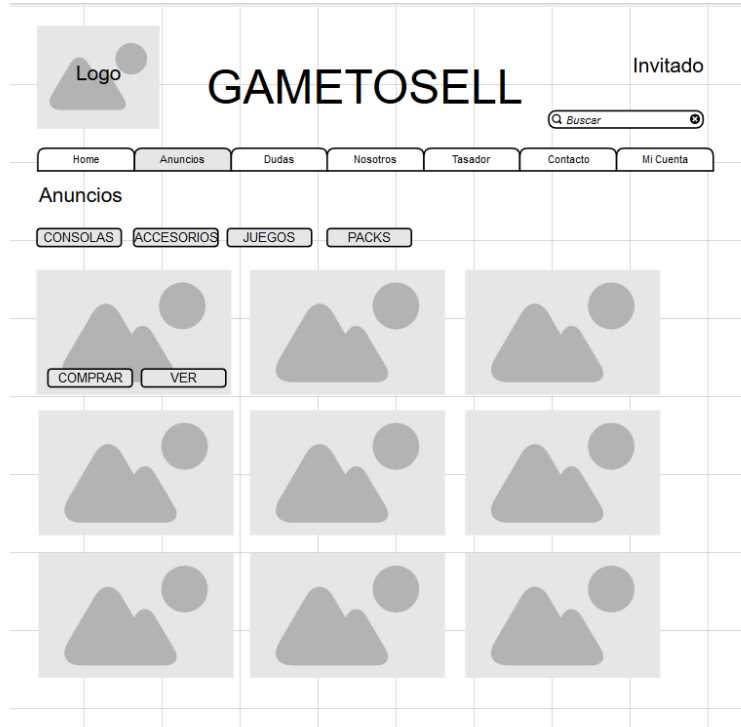
- Home

En esta pantalla el usuario puede ver los anuncios.

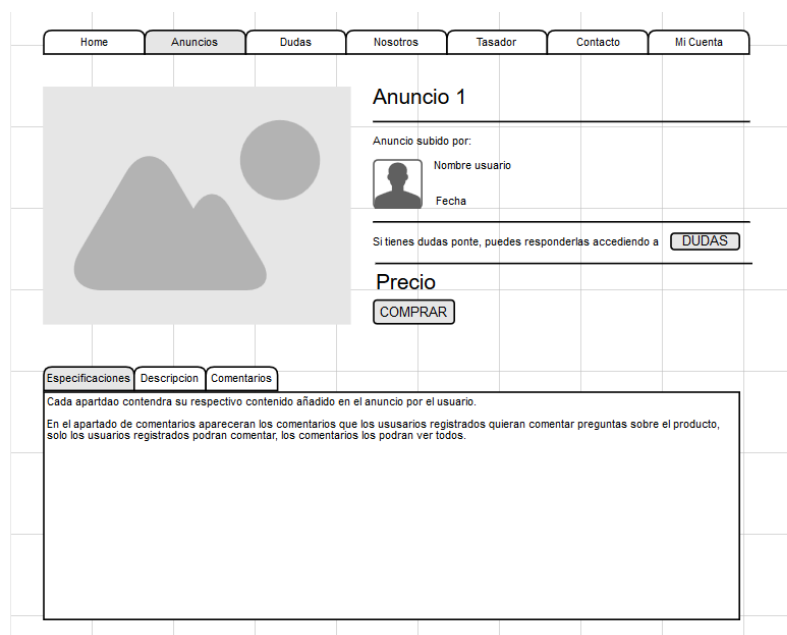


▪ Anuncios

En la opción de anuncios aparecen todos los anuncios disponibles. Dentro de cada anuncio existe la opción de comprar o ver el producto.

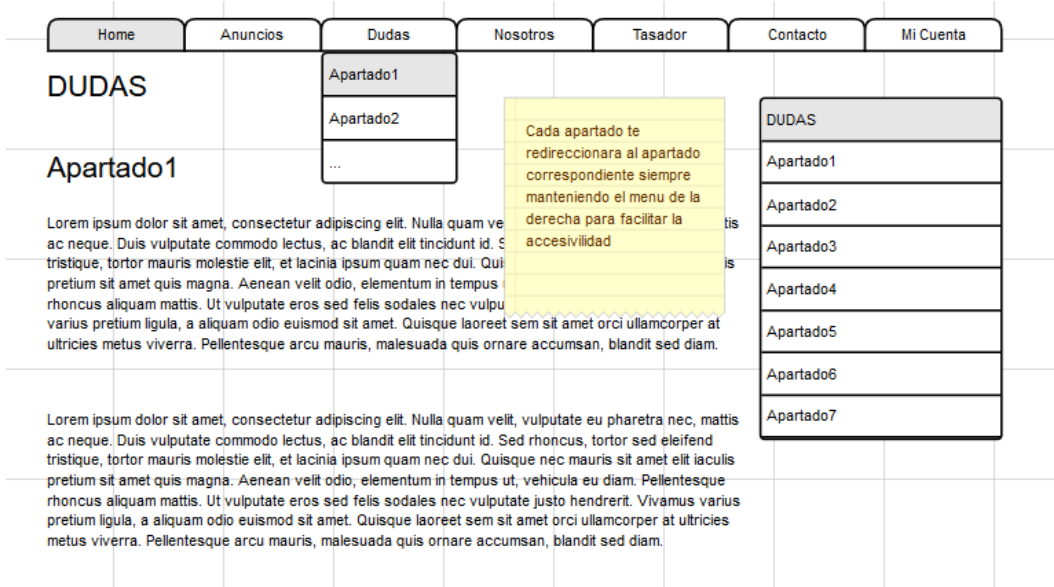


Si seleccionas la opción de ver anuncio aparece la descripción del producto y si al usuario le gusta puede seleccionar la opción de comprar.



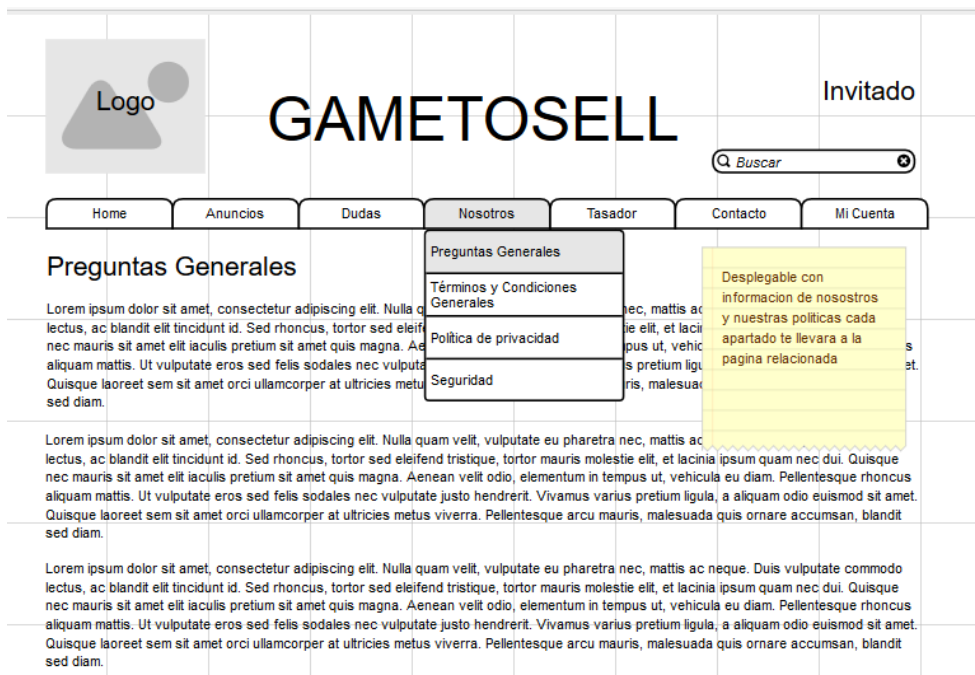
- Dudas

Se explica cualquier inconveniente tanto para el vendedor como para el comprador.



- Nosotros

En este apartado aparecen las preguntas generales, los términos y condiciones generales, la política de privacidad y la seguridad.



- Tasador

Esta es una de las funciones principales que ofrece la página. Desde esta parte el usuario podrá obtener la tasación del artículo que desee, para ello solamente deberá seleccionar las características de su producto. Si el usuario considera que el precio no se adecúa a su valor podrá reportar al administrador, siempre y cuando haya iniciado sesión

Home Anuncios Dudas Nosotros **Tasador** Contacto Mi Cuenta

**TASADOR**

**TIPO**  
CONSOLA  
ACCESORIO  
JUEGO

**PLATAFORMA**  
SONY  
MICROSOFT  
.....

**MODELO**  
PSP  
PS1  
PS2 FAT  
PS2 SLIM

**CAPACIDAD\***  
500GB  
1TB

**CUANTO VALE MI CONSOLA**

LAS CASILLAS QUE VAN SURGIENDO DEPENDERA DE QUE OPCION A ELEGIDO EL USUARIO, POR EJEMPLO SI ELIGUE CONSOLA, SONY, LA SIGUIENTE SERA QUE CONSOLA Y LA SIGUIENTE LAS CARACTERISTICAS

SONY  
PLAY STATION 2 - FAT

PRECIO€ ANUNCIAR

Si crees que el precio no es el adecuado Reportar

UNA VEZ SELECCIONADO TODOS LOS PARAMETROS OBLIGATORIOS APARECERA EL BOTON PARA TASAR LO QUE EL USUARIO QUIERA, CUANDO SE PULSE EL BOTON SALDRA UN DIALOGO MODAL QUE SE ESPECIFICA

- Contacto

Localización de la empresa y dónde se puede enviar cualquier duda.

Home Anuncios Dudas Nosotros Tasador **Contacto** Mi Cuenta

**SITUACION**

**CONTACTO**

EMAIL

ASUNTO

MENSAJE

ENVIAR

- Login

Es necesario iniciar sesión para poder publicar un anuncio o reportar un precio y para poder realizar actividades de compra y venta.

The screenshot shows the top navigation bar of the GAMETOSELL website. On the left is a 'Logo' placeholder. In the center is the site name 'GAMETOSELL'. On the right, it says 'Invitado' and there is a search bar with the text 'Buscar'. Below the navigation bar is a menu with items: Home, Anuncios, Dudas, Nosotros, Tasador, Contacto, and Mi Cuenta. The main content area features a 'Iniciar sesion' form with input fields for 'EMAIL' and 'PASSWORD', an 'INICIAR' button, and a link for 'Olvidaste la contraseña?'. Below this is a section for 'Eres Nuevo?' with a 'CREAR CUENTA' button.

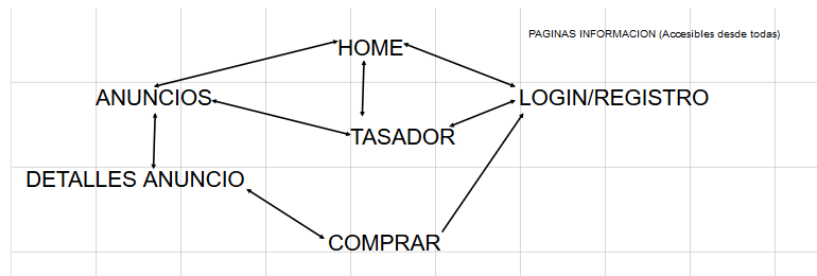
- Registro

El usuario podrá crear una cuenta si cumplimenta los datos correctamente.

The screenshot shows the top navigation bar of the GAMETOSELL website, identical to the previous one. The main content area features a 'Crear Cuenta' form with input fields for 'USUARIO', 'EMAIL', 'PASSWORD', and 'REPETIR PASSWORD', and a 'CREAR' button. Below this is a section for 'Eres Miembro?' with an 'INICIAR SESION' button.

- Wireframe

Desde home podemos acceder a las paginas funciones principales del menú de la cabecera. Además también podrás acceder a tu cuenta mediante login y registro. Una vez seleccionado el anuncio podrás acceder a los detalles del mismo. Y desde esa misma pantalla la aplicación te da la opción de comprar siempre que inicies sesión. Si no lo habías hecho previamente aparecerá una pantalla de login.



## Privado

- Mi Cuenta

Al iniciar sesión el usuario podrá ver sus datos personales y modificarlos si fuera necesario, aquí encontrará las direcciones de envío, la contraseña y la opción de darse de baja.

En la parte derecha de la pantalla se mostrará un menú relacionado con los movimientos efectuados por el usuario registrado, como los artículos publicados por dicho usuario.

The wireframe shows a user profile page with a navigation bar at the top containing: Home, Anuncios, Dudas, Nosotros, Tasador, Contacto, and Mi Cuenta. The main content area is divided into several sections:

- Datos Personales:** Includes a profile picture placeholder, a 'Cambiar' button, and input fields for NOMBRE, APELLIDOS, FECHA NACIMIENTO, and TELEFONO, with a 'CAMBIAR' button.
- Direcciones:** Includes input fields for Direccion1 and Direccion2, 'Cambiar/Eliminar' buttons, and an 'AÑADIR' button.
- Cambiar Password:** Includes input fields for PASSWORD ACTUAL, PASSWORD NUEVO, and REPETIR PASSWORD, with a 'CAMBIAR' button.
- DARSE DE BAJA:** Includes a 'DARSE DE BAJA' button.

On the right side, there is a 'MICUENTA' menu with the following items: Perfil, Mis articulos, Mis Compras, Mis Ventas, and Mis Seguimiento. A yellow callout box contains the text: 'Si pulsamos el boton añadir se abre una ventana modal con los datos necesarios para añadir una nueva direccion. Si los datos son correcto se añadira sino salra mensaje de error.'

- Pasarela De Compra

Una vez que se haya elegido el artículo deseado, se seleccionará la opción de comprar y aparecerá esta página en la que el usuario puede elegir los parámetros que desea, como por ejemplo la dirección de envío o cualquier dato pertinente sobre la compra.

The wireframe shows a 'COMFIRMAR COMPRA' page. On the left, there is a placeholder for an image and a list of product details: 'Titulo', 'DESCRIPCION', and 'PRECIO'. On the right, under 'DATOS DE ENVIO', there are four input fields for 'NOMBRE', 'APELLIDOS', 'FECHA NACIMIENTO', and 'TELEFONO'. Below this, the 'DIRECCION DE ENVIO' section has two radio buttons for 'Direccion1' and 'Direccion2'. The 'FORMAS DE PAGO' section has two radio buttons for 'PAYPAL' and 'CONTRA REEMBOLSO'. At the bottom right, it displays 'GASTOS DE ENVIO: 20€' and 'TOTAL A PAGAR: 120€' in red. A 'REALIZAR EL PAGO' button is centered at the bottom.

- Publicar Anuncio

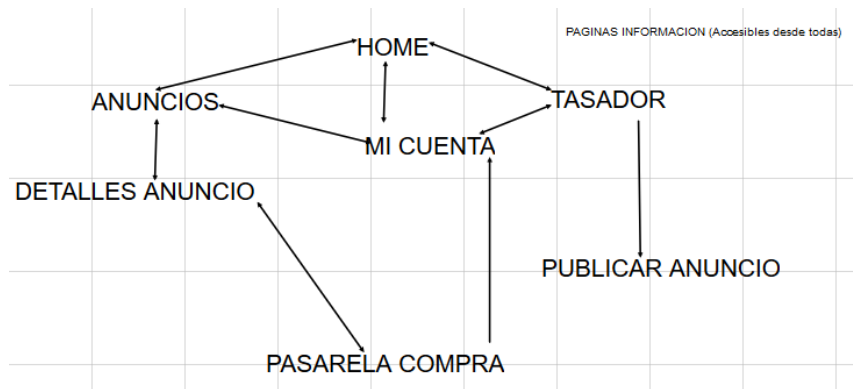
Una vez que el tasador ha realizado su función, se puede publicar el artículo.

The wireframes show two steps of the 'Publicar Anuncio' process. The first step, 'Publicar', includes input fields for 'TITULO' and 'DESCRIPCION', a price selection area with a checked '120€' option and an unchecked 'PON TU EL PRECIO' option, and a 'SIGUIENTE >>' button. The second step, 'IMAGENES', features a 'Principal' label, an 'Examinar' button, a placeholder for an image, and a 'PUBLICAR >>' button. A green callout box at the bottom left explains: 'Al publicar un anuncio previamente se le llevara al tasador para que tase su producto y a continuacion de daremos la opcion de publicar'.

- Wireframe

El usuario que inicia sesión tiene acceso a las mismas partes del menú que en la parte pública, pero a diferencia de ésta, una vez iniciada la sesión puedes entrar en tu perfil de usuario así como tasar productor y posteriormente publicar el anuncio.

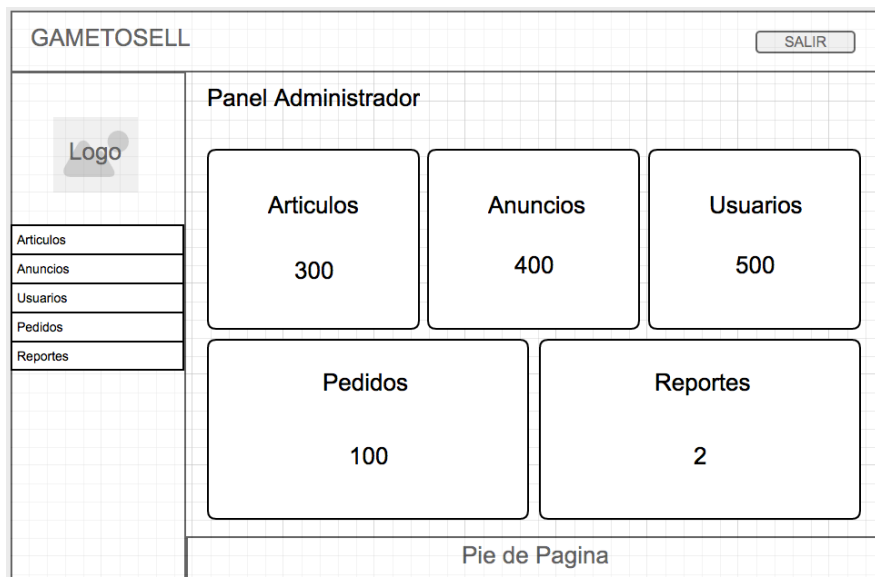
Desde el detalle del anuncio puedes acceder a la compra del producto



## Administrador

- Home

El administrador podrá acceder a todos los apartados a partir del menú de la izquierda





- Anuncios

En anuncios el administrador podrá ver un listado de los anuncios y eliminar los que hayan sido reportados por los usuarios.



GAMETOSELL SALIR

Logo

- Articulos
- Anuncios**
- Usuarios
- Pedidos
- Reportes

En anuncios saldra un desplegable donde podremos elegir entre juegos, consolas, accesorios y todos

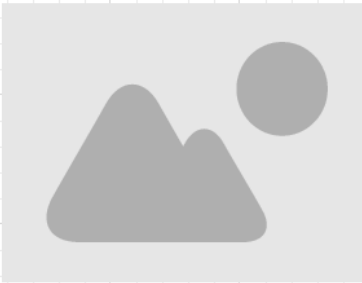
### Listado Anuncios

▼ Titulo	▼ Descripcion	▼ Precio	▼ Usuario	▼ Ver/Editar/Eliminar
nombre1	....	100€	usuario1	 
nombre2	.....	100€	usuario1	
nombre3	...	100€	usuario1	
nombre4	.....	100€	usuario1	

Pie de Pagina

Esto es lo que puede ver el administrados cuando selecciona la opción de ver anuncio.

### Articulo 1



Nombre

Descripcion

Precio

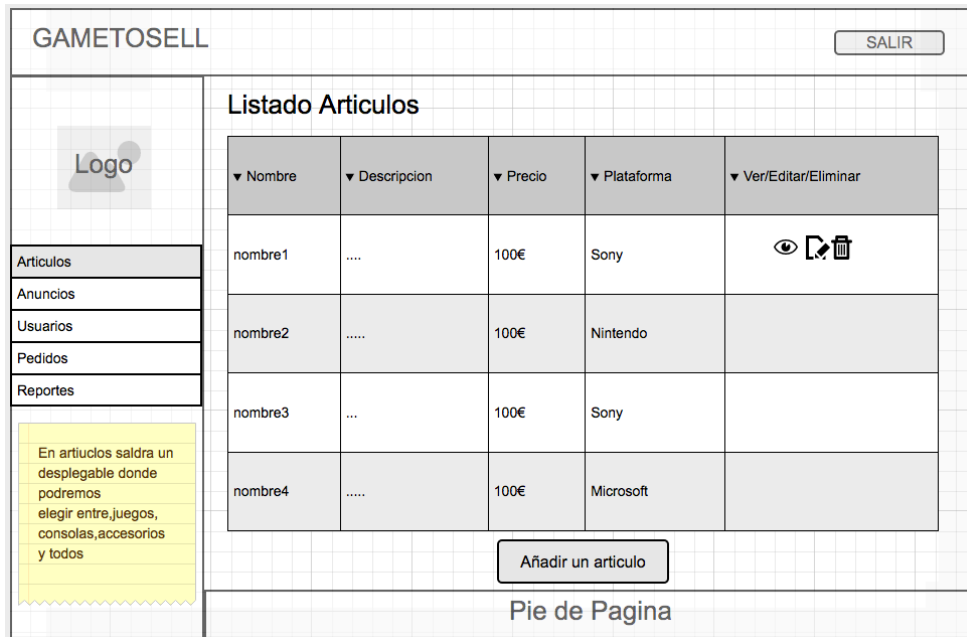
Usuario

Comentarios

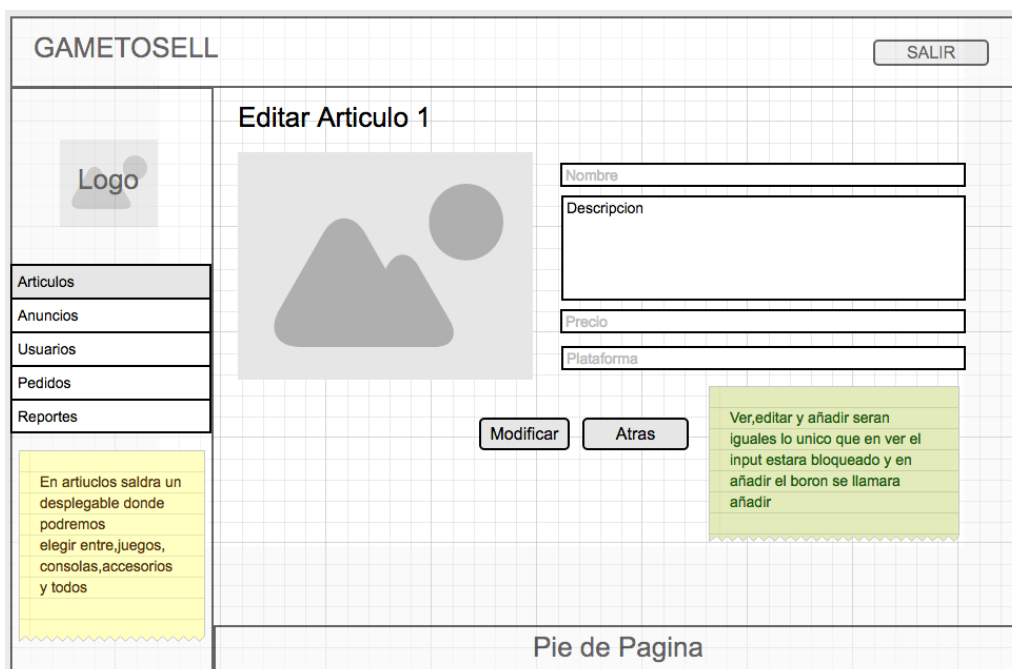
Aceptar Atras

- Artículos

El administrador podrá obtener un listado de todos los artículos disponibles en la base de datos así como obtener una referencia de la tasación de los mismos. Las funciones a las que tiene acceso desde esta parte del menú serían, ver artículo, añadir, editar artículo.




En la siguiente imagen se puede ver como el administrador accede a los datos de cada artículo.





- Usuarios

Dentro de la opción de usuarios el administrador puede obtener una lista de los usuarios registrados y ver los detalles de estos, así como dar de baja a los que hayan producido algún incidente.




GAMETOSELL					
				SALIR	
 Artículos Anuncios <b>Usuarios</b> Pedidos Reportes	<b>Listado Usuarios</b>				
	▼ Nombre	▼ Apellidos	▼ Email	▼ Baja	▼ Ver/DarBaja
	nombre1	....	...	NO	<input type="checkbox"/> <input checked="" type="checkbox"/>
	nombre2	.....	...	NO	
	nombre3	...	...	NO	
nombre4	.....	...	NO		
Pie de Pagina					

A continuación mostramos la pantalla que le aparecerá al administrador para poder modificar los datos de los usuarios registrados.

GAMETOSELL	
	SALIR
 Artículos Anuncios <b>Usuarios</b> Pedidos Reportes	<b>Usuario 1</b>  <input checked="" type="checkbox"/> Baja
	<input type="text"/> Nombre <input type="text"/> Apellidos <input type="text"/> email <input type="text"/> user <input type="text"/> Fecha nacimiento <input type="text"/> telefono <input style="height: 30px;" type="text"/> Direcciones
	<input type="button" value="Aceptar"/> <input type="button" value="Atras"/>
	Pie de Pagina

- Pedidos

Aquí el administrador podrá ver los pedidos realizados, podrá ver un pedido en concreto, modificar el pedido y eliminar.

GAMETOSELL		Listado Pedidos								
<div style="text-align: center;">Logo</div> <table border="1"> <tr><td>Articulos</td></tr> <tr><td>Anuncios</td></tr> <tr><td>Usuarios</td></tr> <tr><td><b>Pedidos</b></td></tr> <tr><td>Reportes</td></tr> </table> <p>En pedidos saldra un desplegable donde podremos elegir entre,juegos, consolas, accesorios y todos</p>	Articulos	Anuncios	Usuarios	<b>Pedidos</b>	Reportes	▼ Anuncio	▼ Comprador	▼ Precio	▼ Vendedor	▼ Ver/Editar/Eliminar
	Articulos									
	Anuncios									
	Usuarios									
	<b>Pedidos</b>									
Reportes										
nombre1	....	100€	usuario1	  						
nombre2	.....	100€	usuario1							
nombre3	...	100€	usuario1							
nombre4	.....	100€	usuario1							
Pie de Pagina										

En la siguiente imagen podemos ver la página que le aparece al administrador para gestionar los datos de información de los pedidos.

GAMETOSELL		Pedido N°						
<div style="text-align: center;">Logo</div> <table border="1"> <tr><td>Articulos</td></tr> <tr><td>Anuncios</td></tr> <tr><td>Usuarios</td></tr> <tr><td><b>Pedidos</b></td></tr> <tr><td>Reportes</td></tr> </table> <p>En pedidos saldra un desplegable donde podremos elegir entre,juegos, consolas, accesorios y todos</p>	Articulos	Anuncios	Usuarios	<b>Pedidos</b>	Reportes		Nombre	<input type="text"/>
	Articulos							
	Anuncios							
	Usuarios							
	<b>Pedidos</b>							
Reportes								
Descripcion	<input type="text"/>							
Precio	<input type="text"/>							
Usuario	<input type="text"/>							
Estado	<input type="text"/>							
		<input type="button" value="Aceptar"/> <input type="button" value="Atras"/>						
Pie de Pagina								

- Reportes

Aquí el administrador podrá ver los reportes que han realizado los usuarios registrados.

**GAMETOSELL** SALIR

Logo

### Listado Reportes Anuncio

▼ Anuncio	▼ Reportador	▼ Reportado	▼ Estado	▼ Revisar
nombre1	....	...	revisado	<input checked="" type="checkbox"/>
nombre2	.....	....	No revisado	<input type="checkbox"/>
nombre3	...	....	revisado	<input type="checkbox"/>
nombre4	.....	....	revisado	<input type="checkbox"/>

Pie de Pagina

En reportes saldra un desplegable donde podremos elegir entre reportes de precio y reportes de anuncios

Si seleccionamos revisar en reporte precio aparecerá esta página, el administrador podrá modificar los campos sin la situación lo requiere.

**GAMETOSELL** SALIR

Logo

### Reporte Precio Artículo 1

Nombre

Descripción

Precio

Plataforma

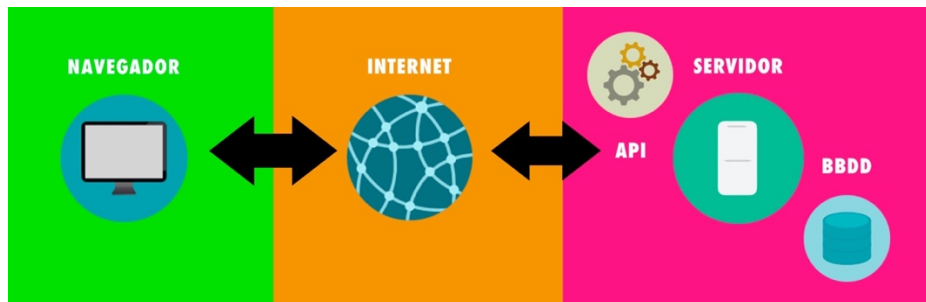
Pie de Pagina

En reportes saldra un desplegable donde podremos elegir entre reportes de precio y reportes de anuncios

# IMPLEMENTACIÓN

## Implementación API

Se ha desarrollado un API en php con el objetivo de simplificar el acceso a los procedimientos almacenados en base de datos desde las diferentes aplicaciones (Android, IOS, Web).



Partes de las que se compone el sistema:

- Cliente: que será el navegador, se conecta a internet.
- Internet: que se comunica con el servidor.
- Servidor: Dónde se encuentran alojados el API y la Base De Datos(BBDD). La función del servidor es recibir la petición del cliente y a partir de ahí accede al API, en ese momento el API ya se encarga de mandar la respuesta a internet que lo envía al navegador.

Cada servicio del API opera sobre un procedimiento almacenado, el resultado de la ejecución de los procedimientos será enviado a internet y este lo devolverá al navegador en formato JSON.

Para establecer la comunicación entre el API y la aplicación, se ha creado una función que llamaremos callAPI. Esta función favorecerá la entrada de una respuesta que contenga el resultado de la petición. Todas las respuestas tienen dos partes, el resultado y el mensaje.

El resultado sirve para saber si la petición a tenido éxito (indicado con un 1). También para indicar que no devuelve ningún resultado o faltan parámetros en la petición

se indicará con el valor de 0. Si el mensaje de respuesta existe y da un resultado con éxito, se recogerá el mensaje y se tratará según lo debido.

El fichero principal del API es “conexión\_bd.php” que implementa una clase de nombre “Base de Datos” que contiene varias funciones:

La función “conectar”, es la que se encarga de realizar la conexión con la base de datos, basándose en las credenciales que estén establecidas en el fichero “login\_bd.php”:

```
public function conectar(){
    $this->conexion = mysql_connect(BD_HOST, BD_USER, BD_PASS);

    if ($this->conexion == 0) DIE("Lo sentimos, no se ha podido conectar con MySQL: " . mysql_error());
    $this->db = mysql_select_db(BD_NAME, $this->conexion);

    if ($this->db == 0) DIE("Lo sentimos, no se ha podido conectar con la base datos: " . BD_NAME);

    mysql_set_charset('utf8',$this->conexion);

    return true;
}
```

La función “desconectar”, su único objetivo es cerrar la conexión con la base de datos:

```
public function desconectar(){
    if ($this->conectar->conexion) {
        mysql_close($this->$conexion);
    }
}
```

La función “faltanParametros”, se ejecutará cuando se intente realizar una petición y el número de parámetros sean incorrectos:

```
public function faltanParametros(){
    $response["success"]=0;
    $response["message"]= "Faltan parámetros";
    echo preg_replace("/\\\\\\u{[a-f0-9]{4}}/e", "iconv('UCS-4LE','UTF-8',pack('V', hexdec('U$1')))", json_encode($response));
}
```

Por último, la función “ejecutarProcedure” es la encargada de ejecutar el procedimiento almacenado en la base de datos:

```
public function ejecutarProcedure($procedimiento){
    //echo($procedimiento);
    $resultset = mysql_query('CALL '.$procedimiento);
    $response = array();

    if (mysql_num_rows($resultset) > 0){
        $response["success"]=1;
        while($r = mysql_fetch_assoc($resultset)){
            $response["message"][] = $r;
        }
    }else{
        $response["success"]=0;
        $response["message"]= "Ningún resultado";
    }

    echo preg_replace("/\\\\\\\\u{[a-f0-9]{4}}/e", "iconv('UCS-4LE','UTF-8',pack('V', hexdec('U$1')))", json_encode($response));
}
```

El resto de ficheros tienen un comportamiento similar, solo varían el procedimiento que ejecutan y los parámetros de los mismo, por ejemplo vamos a ver cómo sería el procedimiento de borrar un usuario:

```
<?php
require_once "conexion_bd.php";

$db = new BaseDatos();

if($db->conectar()){

    if(isset($_GET['p_id'])){
        $id = $_GET['p_id'];
        $db->ejecutarProcedure("Sp_BorrarUsuario(".$id.")");
    }else{
        $db->faltanParametros();
    }

    $db->desconectar();
}
?>
```

Como se puede observar, la lógica es la misma para todos. Primero se intenta realizar la conexión a la base de datos, si es posible realizar la conexión a la base de datos, se comprobará que se está llamando con los parámetros adecuados. En caso contrario se enviará la respuesta de que faltan parámetros y se cerrará la conexión en base de datos. Si por el contrario son correctos los parámetros se serán llamados al procedimiento almacenado en base de datos para que intente realizar la acción acordada.

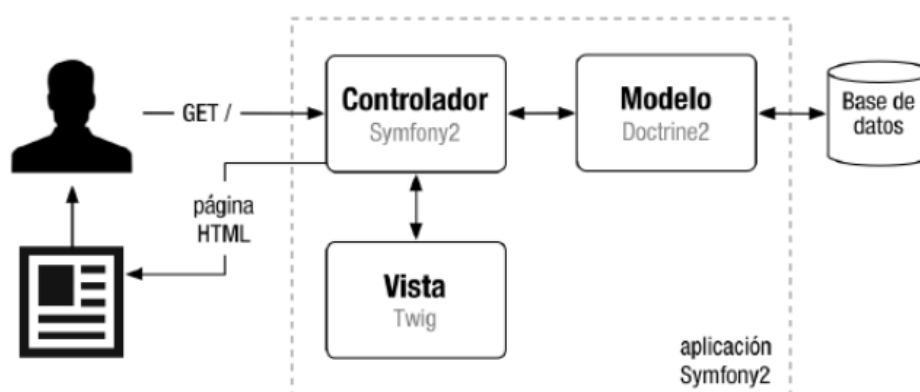


## Implementación Web

### Modelo Vista Controlador

Es un patrón o modelo de abstracción de desarrollo de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos.

- **Modelo:** representa el dominio de los datos. Define la lógica de negocio (la base de datos pertenece a esta capa).
- **Vista:** es la interfaz de usuario.
- **Controlador:** es el responsable de ejecutar las peticiones del usuario. Es el que invoca peticiones al modelo y llama a las vista.



Cuando un usuario solicita una página de la aplicación web, internamente sucede lo siguiente:

1. El sistema de enrutamiento determina qué controlador está asociado con la página solicitada.
2. Symfony2 ejecuta el controlador asociado a la portada. Un controlador no es más que una clase PHP en la que puedes ejecutar cualquier código que quieras.
3. El controlador solicita al modelo los datos que se requieren. El modelo no es más que una clase PHP especializada en obtener información, normalmente de una base de datos.

4. Con los datos devueltos por el modelo, el controlador solicita a la vista que cree una página mediante una plantilla y que inserte los datos del Modelo.
5. El Controlador entrega al servidor la página creada por la Vista.

A pesar de que puedes llegar a hacer cosas muy complejas con Symfony2, el funcionamiento interno siempre es el mismo:

1. El controlador manda y ordena
2. El modelo busca la información que se le pide
3. La vista crea páginas con plantillas y datos.

## Empezando Symfony

Antes de empezar hay que crear un *bundle* ("paquete"), todo el código de la aplicación se almacena dentro de bundles\*.

Ejecutamos el comando:

```
$ php app/console generate:bundle --namespace=Acme/G2SBundle --format=yml
```

Se ha creado un directorio para el bundle en `src/Acme/G2SBundle`

La estructura es esta:

```
game2sell/  
  app/  
    cache/  
    config/  
    logs/  
    Resources/  
  bin/  
  src/  
    Acme/  
      G2SBundle/  
        Controller/  
        Resources/  
        ...  
  vendor/  
    symfony/  
    doctrine/  
    ...  
  web/  
    app.php  
    ...
```

“app” y “bin” son carpetas internas de symfony.

Después se encamina la petición al código que la maneja tratando de hacer coincidir la *URL* solicitada contra algunas rutas configuradas:

```
# src/Acme/Resouces/config/routing.yml
home:
  path:      /home
  defaults:  {_controller: AcmeG2SBundle:Home:index}

contacto:
  path:      /contacto
  defaults:  {_controller: AcmeG2SBundle:Contacto:index}

# ...
```

Cuando solicites `home`, el controlador `AcmeG2SBundle:Home:index` será ejecutado.

*\*Un bundle no es más que un directorio que almacena todo lo relacionado con una función específica, incluyendo clases PHP, configuración, e incluso hojas de estilo y archivos de JavaScript.*

## Controladores

Función o método *PHP* que se encarga de las *peticiones* entrantes y devuelve las *respuestas* (a menudo código *HTML*).

*Symfony2* elige el controlador basándose en el valor del `_controller` de la configuración de ruta: `AcmeG2SBundle:Home:index`. Esta cadena es el nombre lógico del *controlador*, y hace referencia al método `indexAction` de la clase `Acme\G2Sbundle\Controller\HomeController`:

```
// src/Acme/G2SBundle/Controller/HomeController.php
namespace Acme\G2SBundle\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\Controller;

class HomeController extends Controller
{
```

```

    public function indexAction()
    {
        return $this->render('AcmeG2SBundle:Web:home.html.twig');
    }
}

```

El nombre de la plantilla, `Acme G2SBundle:Web:home.html.twig`, es el *nombre lógico* de la plantilla y hace referencia al archivo `Resources/views/Web/home.html.twig` dentro del *AcmeG2SBundle* (ubicado en `src/Acme/ G2SBundle`).

## Vistas

Se ha utilizado Twig como motor de plantillas. El controlador procesa la plantilla `AcmeG2SBundle:Web:home.html.twig`:

```

{# src/Acme/G2SBundle/Resources/views/Web/home.html.twig #}
{% extends 'AcmeG2SBundle::base.html.twig' %}

<title>GameToSell - {% block title %}Inicio{% endblock
%}</title>

{% block body %}
<section class="revolution_slider">
    <#...
{% endblock %}

```

Se utiliza el `extends` para mostrar lo contenido dentro del fichero “`base.html.twig`”, este `extends` se ha utilizado durante el desarrollo del proyecto para realizar la cabecera y el pie, tratando de no repetir constantemente el mismo código.

## Modelo

**Mapeo de objeto-relacional**, es un modelo de programación que consiste en la transformación de los datos, en una serie de entidades que simplifiquen las tareas básicas de acceso a los mismos.

En la aplicación se han creado diferentes entidades. Por ejemplo Anuncio:

Objeto `Anuncio` representa a los anuncios. Crea esta clase dentro del directorio `Entity` del *bundle* `AcmeG2SBundle`:

Esta clase es la "entidad" o "entity", esto implica que es una clase muy sencilla y sólo se utiliza para almacenar datos. Aunque se trata de una clase muy básica, cumple su objetivo de representar a los anuncios en la aplicación.

Ejecutamos el comando:

```
$ php app/console doctrine:generate:entity
```

Esto genera el código en la aplicación:

```
// src/Acme/G2SBundle/Entity/Anuncio.php
namespace Acme\StoreBundle\Entity;

class anuncio
{
    private $id;

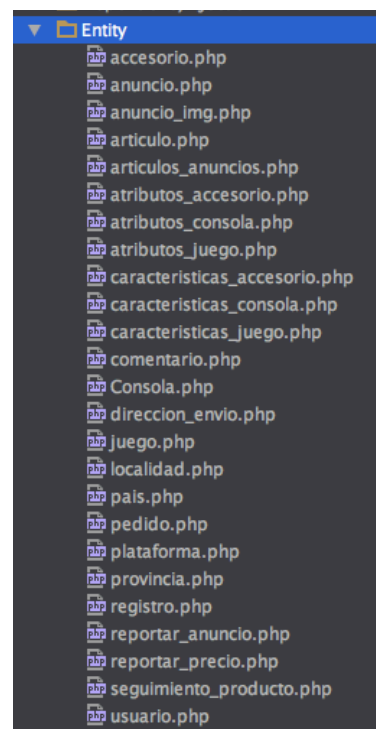
    private $titulo;

    private $precio;
}
```

Como `Anuncio` es una clase PHP, es necesario crear métodos *getters* y *setters* (`getId()`, `setId()`, etc.) para poder acceder a sus propiedades.

Ejecutamos el comando para que se añadan automáticamente:

```
$ php app/console doctrine:generate:entities
Acme/G2SBundle/Entity/Anuncio
```



## Formularios

*Symfony2* integra un componente `Form` que se ocupa de facilitarnos la utilización de formularios.

Los usuarios publicaran anuncios en la aplicación, hay que crear un formulario para poder hacerlo. Para realizar el formulario hay que tener realizada la entidad `Anuncio` que representa y almacena los datos para un anuncio.

Se construye el formulario en un fichero separado:

```
// src/Acme/G2SBundle/Form/AnuncioFile.php
namespace Acme\G2SBundle\Form;

use Symfony\Component\Form\AbstractType;
use Symfony\Component\Form\FormBuilderInterface;

class AnuncioFile extends AbstractType
{
    public function buildForm(FormBuilderInterface $builder,
array $options)
    {
        $builder
            ->add('titulo', 'text')
            ->add('descripcion', 'textarea')
            ->add('precio', 'text');
    }

    public function configureOptions(OptionsResolver $resolver)
    {
        $resolver->setDefaults(array(
            'data_class' => 'Acme\G2SBundle\Entity\anuncio',
        ));
    }

    public function getName()
    {
        return 'anuncio';
    }
}
```

Esta nueva clase contiene todas las indicaciones necesarias para crear el formulario de anuncio (el método `getName()` devolverá un identificador único para este «tipo» de formulario).

## Controlador:

```
// src/Acme/G2SBundle/Controller;
use Symfony\Component\HttpFoundation\Request;
use Acme\G2SBundle\Entity\anuncio;

public function detalleAction(Request $request)
{
    //Form anuncio
    $anuncio = new anuncio();
    $form = $this->createForm(new AnuncioFile(), $anuncio);
    $form->handleRequest($request);

    //Se comprueba si ha sido enviado
    if ($form->isSubmitted()) {
        ...
    }

    //Se envía a la plantilla
    return $this-
>render('AcmeG2SBundle:Web:detalle_articulo.html.twig', array(form =>
    $$form->createView()));
```

Desde el controlador se pasa un objeto `view` especial para formularios a la plantilla:

```
{# src/Acme/G2SBundle/Resources/views/Web/.html.twig #}

{{ form_start(form, {'action': path('home')}) }}
{{ form_widget(form.titulo) }}

<input type="submit" />
{{ form_end(form) }}
```

# Implementación imágenes

## Rol Público

**GAMETOSELL** INICIO ANUNCIOS ¿DUDAS? NOSOTROS TASADOR CONTACTO CUENTA

# GAMETOSELL

EMPIEZA A DISFRUTAR

100% Satisfacción GARANTIZADA

ENVIO EN 24 HORAS

GRANDES PRODUCTOS COMPRA YA!

### ANUNCIOS

TODOS CONSOLAS ACCESORIOS JUEGOS

- Vendo Nintendo 64  
100.00 €  
Comprar
- Vendo DreamCast  
70.00 €  
Comprar
- Oferta N64  
88.00 €  
Comprar
- Dc Vendo  
45.00 €  
Comprar

TASADOR! **DESCUBRE SU PRECIO** EMPEZAR!

**VENDE** PUBLICA TU PROPIO ANUNCIO PUBLICAR!

### PLATAFORMAS

MAGNAVOX SMART. VERY SMART. QUALCOMM Panasonic SNK NEC CASIO

Nosotros: Somos estudiantes, los cuales vimos una oportunidad de negocio y de poder ayudar a mucha gente con el mismo tipo de problema que tenemos hoy en día. Cuanto vale? Por cuanto lo vendo?

Servicios: Mi Cuenta, Crear Cuenta, Deseados, Plataformas

Nosotros: Preguntas Generales, Seguridad, Política de privacidad, Términos y condiciones


Anuncios: Consolas, Accesorios, Juegos, Packs

Contacto: Carretera San Vicente del Raspeig, Sant Vicent del Raspeig, Alicante 03690. 965 90 34 00. solutionseasylife@gmail.com

Copyright © 2015-2016 EasyLifeSolutions. Todos los derechos reservados.


PayPal VISA MasterCard American Express



**GAMETOSSELL** INICIO **ANUNCIOS** ¿DUDAS? NOSOTROS TASADOR CONTACTO CUENTA 


Inicio > Anuncios

### ANUNCIOS




Venta de Juegos Antiguos  
130.00

[Comprar](#)




Vendo Nintendo 64  
100.00

[Comprar](#)




Vendo DreamCast  
70.00

[Comprar](#)



Oferta N64  
88.00



[Comprar](#)



APP MÓVIL

## FRUTA EN MÓVIL

Descargala YA!

**Iniciar Sesión** ✕

Usuario

Contraseña

[Entrar](#)




Nuevo Usuario? [Crea una Cuenta](#)

100% SATISFACCIÓN

ENVIO


GRANDES PRODUCTOS

## Rol Privado

**GAMETOSELL** INICIO ANUNCIOS ¿DUDAS? NOSOTROS TASADOR CONTACTO   

Inicio > Mi Cuenta

### DATOS PERSONALES



**BUSCAR** **SUBIR**

Nombre	<input type="text" value="Blanca"/>
Apellidos	<input type="text" value="Resco"/>
Email	<input type="text" value="blanca@gmail.com"/>
Fecha Nacimiento	<input type="text" value="1991-01-20"/>
Telefono	<input type="text" value="696969696"/>

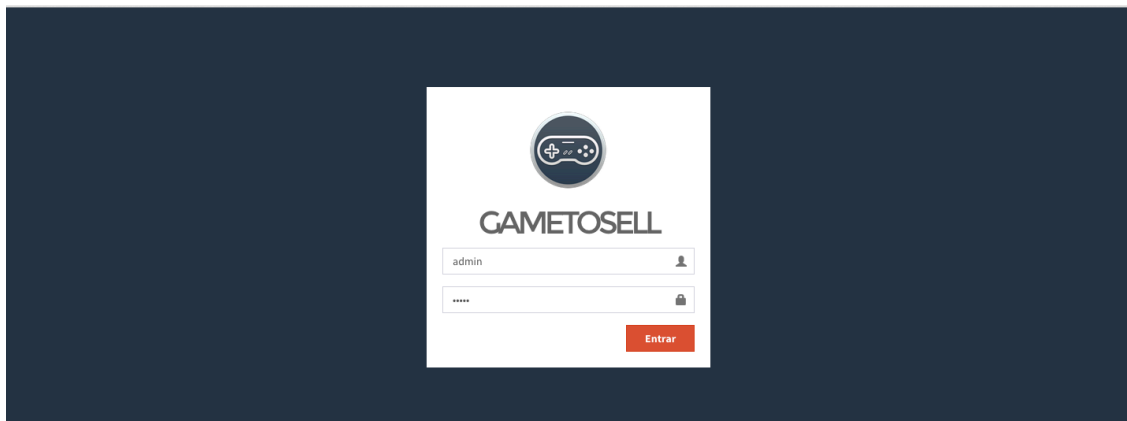
**Guardar**


**Mi Cuenta**

**Perfil**

**Mis Anuncios**

## Rol Administrador



**GAMETOSELL**  Administrador

**Administrador**  
En Línea

**PANEL NAVEGACION**

- Articulos
- Anuncios
- Usuarios
- Pedidos
- Reportes

**Panel** Administrador

**150**  
Articulos  
[Más info](#)

**53%**  
Anuncios  
[Más info](#)

**44**  
Usuarios  
[Más info](#)

**65**  
Pedidos  
[Más info](#)

**65**  
Reportes  
[Más info](#)

Copyright © 2015-2016 EasyLifeSolutions. Todos los derechos reservados

Nombre	Descripción	Precio	Tipo	Ver/Editar
1 PlayStation	La primera consola de Sony.	20.00	Consola	[Ver] [Editar]
2 Final Fantasy VII	Final Fantasy VII es un videojuego de rol desarrollado y publicado por la empresa Square para la plataforma PlayStation.	120.00	Juego	[Ver] [Editar]
3 DualShock	Mando de control analógico de Playstation.	5.00	Accesorio	[Ver] [Editar]
6 Nintendo 64	La quinta videoconsola de sobremesa de Nintendo.	30.00	Consola	[Ver] [Editar]
8 Super Mario 64	El juego más vendido de la N64	45.00	Juego	[Ver] [Editar]
9 Zelda Ocarina Of Time	Primer zelda en 3d, considerado uno de los mejores juegos de la historia.	120.00	Juego	[Ver] [Editar]
10 Dreamcast	La última consola de Sega.	45.00	Consola	[Ver] [Editar]
12 Mando N64	Mando para la Nintendo 64	6.00	Accesorio	[Ver] [Editar]
13 Pistola Logic3	Pistola compatible con la psx, ps1 y ps2.	3.00	Accesorio	[Ver] [Editar]
14 Volante Ferrari	Volante F360 Modena	15.00	Accesorio	[Ver] [Editar]

## SEGURIDAD

### *Problemas principales en la programación de sistemas web*

Gran parte de los problemas de seguridad en las aplicaciones web son causados por la falta de seguimiento en las entradas y salidas del sistema.

Habría que considerar también la exposición accidental de datos que pueden ser empleados en un posible ataque sobre el sistema. Los mensajes de error enviados por el servidor suelen ser de gran utilidad durante el proceso de desarrollo de la aplicación.

Pueden ser empleados maliciosamente cuando siguen apareciendo en un entorno de producción, por lo que es necesario deshabilitar todos estos mensajes y editar algunos, otros también pueden ser utilizados por los atacantes para obtener información sobre nuestro sistema.

### *Prácticas básicas de seguridad web*

#### *Balancear riesgo y usabilidad*

Si bien la usabilidad y la seguridad en una aplicación web no son excluyentes la una de la otra, alguna medida tomada para incrementar la seguridad con frecuencia afecta a la usabilidad.

Es conveniente emplear medidas de seguridad que sean transparentes a los usuarios y que no resulten engorrosas en su empleo. Por ejemplo, el uso de un *login* que solicita el nombre de usuario y contraseña, permite controlar el acceso de los usuarios hacia secciones restringidas de la aplicación.

### *Rastrear el paso de los datos*

Existen funciones globales en la aplicación que sirven para identificar de forma clara las entradas proporcionadas por el usuario. En el caso de nuestra web se ha utilizado *Session*, mediante la cual se almacena el ID del usuario y con esto se puede seguir el rastro durante su recorrido en la aplicación.

### *Filtrar entradas*

Proceso por el cual se prueba la validez de los datos. Mediante el cual nos aseguramos de que los datos son filtrados correctamente al entrar, podemos eliminar el riesgo de que sean utilizados datos contaminados al entrar y esto pueda provocar funcionamientos no deseados en la aplicación.

En el caso de nuestra aplicación los datos son filtrados y comprobados en la base de datos mediante una seguridad que se explica a continuación.

### *Escapado de salidas*

Otro punto importante de la seguridad es el proceso de escapado y su contraparte para codificar o decodificar caracteres especiales de tal forma que su significado original sea preservado. El proceso de escapado debe adecuarse al tipo de salida de que se trate (si es al cliente, a la base de datos, etc.).

### *Métodos empleados para mejorar la seguridad*

Estos son los métodos que se han utilizado en el desarrollo del proyecto con el objetivo de mejorar la seguridad:

### *Contraseña del usuario*

A la hora de almacenar las contraseñas en base de datos, lo que vamos a hacer es aplicar una función de resumen SHA-512. El motivo por el cual no almacenamos la

contraseñas en textos plano es sencillo, porque si almacenamos la contraseña en texto plano cualquier atacante fácilmente podría conseguir las credenciales de acceso.

### Ataques de inyección SQL

Para evitar los ataques mediante inyección de SQL se filtrarán las entradas de datos para evitar los caracteres “especiales” que permiten realizar dichos ataques.

Es muy importante realizar el filtrado en todas las posibles entradas de textos porque un ataque de inyección SQL puede suponer una gran pérdida de datos.

### Protección CSRF

*CSRF* (Cross-site request forgery) o Falsificación de petición en sitios cruzados, es un método por el cual un usuario malintencionado intenta hacer que los usuarios, sin saberlo, envíen datos que no tienen la intención de enviar. Afortunadamente, los ataques *CSRF* se pueden prevenir usando un elemento *CSRF* dentro de tus formularios.

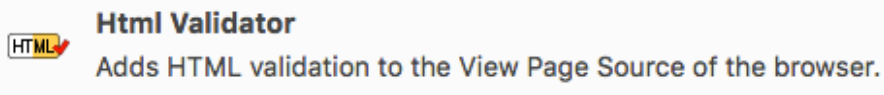
La protección *CSRF* funciona añadiendo un campo oculto al formulario denominado `_token` el cual contiene un valor que sólo tú y tu usuario conocen. Esto garantiza que el usuario y no alguna otra entidad es el que presenta dichos datos. *Symfony* automáticamente valida la presencia y exactitud de este elemento.

## PRUEBAS

Una vez finalizado el desarrollo de la aplicación web se han diseñado una serie de pruebas, para verificar el funcionamiento correcto según los estándares marcados por el HTML.

Al encontrarse el servidor alojado en el ordenador en lugar de tenerlo alojado en internet, las pruebas no son las mismas. Como consecuencia de esto, no se pueden hacer pruebas de carga, ni tampoco se puede validar el estándar HTML mediante el W3c Online Validation.

Por ello se ha utilizado una extensión en el navegador Firefox que permite hacer las mismas comprobaciones y detecta los mismos errores.



Se realiza la prueba con el fichero home.php, la dirección es localhost:8000/home

Type	Line	Column	HTML errors and warnings
Error			The document content type is not HTML : undefined
Result			0 errors / 0 warnings
Info			W3c Online Validation
Info	84	41	The "banner" role is unnecessary for element "header".
Info	126	110	Element "nav" does not need a "role" attribute.
Info	88	36	Section lacks heading. Consider using "h2"- "h6" elements to add identi...
Info	758	69	Section lacks heading. Consider using "h2"- "h6" elements to add identi...
Info	7	16	This document appears to be written in Spanish but the "html" start tag ...

Se puede comprobar que la pagina localhost:8000/home pasaría el W3 sin problemas,

Se ha realizado el test con todas las páginas y ha respondido con éxito. Si la web se subiera al servidor de internet y se comprobara el test online, seguro que estaría correcto.

## PROBLEMAS Y DIFICULTADES

### *Dificultad*

La dificultad principal que se ha detectado ha sido el propio framework, y las variaciones que presenta con el lenguaje PHP. Tras haber trabajado varias semanas con el framework se ha ido consiguiendo conocer más profundamente sus características y esto ha facilitado el desarrollo posterior de la aplicación web.

### *Problemas*

El principal problema que se ha detectado ha sido el envío de los formularios, debido a que es totalmente diferente al que se utiliza con el lenguaje PHP.

En Symfony se debe crear un fichero separado donde se añaden los campos del formulario, asignándole a su vez el campo de un objeto previo, que se encuentra en la

carpeta de una entidad. Posteriormente en el controlador se crea un nuevo objeto del mismo tipo que en el formulario. Y a partir de ahí ya se ejecuta en el fichero que estaba separado, lugar en el que se cargaran los datos enviados y donde luego serán obtenidos.

Otra dificultad que se ha presentado a lo largo del desarrollo de la web han sido las plantillas Twig, debido a que utilizan su propio lenguaje y esto ha creado una dificultad añadida. Para ello se realizó una búsqueda bibliográfica tratando de encontrar solución a dicho problema. La propia página oficial de Twig ha sido imprescindible para el conocimiento de dicha tarea.

## CONCLUSIONES

Tras realizar el desarrollo de la aplicación web y haber aprendido a utilizar las herramientas y frameworks, se puede decir que todo esto ha sido un camino duro pero de mucho aprendizaje, por ello estoy satisfecho con el resultado.

De haber tenido un mayor conocimiento de sobre Symfony y Twig el proceso hubiera sido más sencillo, pero entonces no hubiese aprendido de manera autónoma a utilizar ambas herramientas.

Al haber realizado la base de datos y el API en el servidor junto con otros 2 compañeros esta tarea ha supuesto un extra de comunicación y coordinación entre nosotros. Se podría decir que al principio nos costó adaptarnos a realizar las tareas en mesa de 3. Pero posteriormente utilizamos la metodología Trello para comunicarnos y unificar las tareas comunes.

Creo que los objetivos marcados al comienzo de este proyecto se han conseguido con éxito, habiendo desarrollado la aplicación web que se buscaba. Por ello cabe decir que el trabajo común con mis compañeros y la parte individual de la aplicación web ha salido como se esperaba.

Para finalizar cabe mencionar el gran trabajo realizado por los 3 compañeros que decidimos arriesgarnos a realizar un proyecto de gran envergadura que ofreciera el servicio de GAME2SELL desde diferentes dispositivos, cubriendo la parte WEB, aplicación ANDROID e IOS.

### *Mejoras y ampliaciones*

Los proyectos se pueden alargar el tiempo que quieras, debido a que las tecnologías y la sociedad evoluciona rápido, por eso siempre pueden haber mejoras y ampliaciones:

- Utilizar parte de la base de datos SQL a una base de datos MongoDB (no SQL), y así mejorar la velocidad a la hora de realizar las consultas.
- Desarrollar más el proceso de tasación, poniendo atributos a los artículos como el color, formato, etc. Esto podría hacer que aumente el valor del artículo en el mercado.
- Añadir una cadena (Salt) en la tabla usuario, que sirva para mejorar la seguridad del almacenamiento de las contraseñas. Si añadimos una cadena estamos eliminando la posibilidad de que el resultado pueda buscarse a partir de una lista de pre calculados de hash.
- Poder iniciar cuenta con Google, Facebook, etc. Para que al usuario no le cueste ningún esfuerzo y pueda registrarse en la web con más facilidad, para así poder aumentar el número de usuarios.
- Realizar la aplicación internacional, es decir que este en diferentes idiomas para poder llegar a todo el mundo, tanto la aplicación como la base de datos están preparados para ese cambio.
- Subir la aplicación a un servidor en internet, para poder ofrecer el servicio a todo aquel que esté interesado en las transacciones de compra y venta de videojuegos o videoconsolas.



## BIBLIOGRAFÍAS Y REFERENCIAS

contributors, p. (2016). *phpMyAdmin*. [online] phpMyAdmin. Disponible en: <https://www.phpmyadmin.net/> [Visitada 10 Sep. 2016].

Es.stackoverflow.com. (2016). *Stack Overflow en español*. [online] Disponible en: <http://es.stackoverflow.com/> [Visitada 10 Sep. 2016].

Librosweb.es. (2016). *Symfony 2.4, el libro oficial*. [online] Disponible en: [http://librosweb.es/libro/symfony\\_2\\_4/](http://librosweb.es/libro/symfony_2_4/) [Visitada 3 Mar. 2016].

Moqups.com. (2016). *Online Mockup, Wireframe & UI Prototyping Tool · Moqups*. [online] Disponible en: <https://moqups.com/> [Visitada 10 Sep. 2016].

Stackoverflow.com. (2016). *Stack Overflow*. [online] Disponible en: <http://stackoverflow.com/> [Visitada 10 Sep. 2016].

Symfony.com. (2016). *Symfony 3.1 Documentation*. [online] Disponible en: <http://symfony.com/doc/current/index.html> [Visitada 10 Mar. 2016].

Trello.com. (2016). *Trello*. [online] Disponible en: <https://trello.com/> [Visitada 5 Abril. 2016].

Twig.sensiolabs.org. (2016). *Documentation - Twig - The flexible, fast, and secure PHP template engine*. [online] Disponible en: <http://twig.sensiolabs.org/documentation> [Visitada 10 Abril. 2016].

# ANEXO

## Anexo 1

### Diccionario de datos

#### ACCESORIO

Columna	Tipo	Nulo	Predeterminado	Comentarios
id	int(11)	No		
id_consola	int(11)	No		

#### ANUNCIO

Comentarios de la tabla: almacena los anuncios de los usuarios

Columna	Tipo	Nulo	Predeterminado	Comentarios
id	int(11)	No		identifica el anuncio
titulo	varchar(100)	No		identifica el titulo del anuncio
descripcion	varchar(1000)	No		descripcion del anuncio
precio	float(10,2)	No		precio puesto por el usuario del anuncio
id_usuario	int(11)	No		
activo	int(1)	No	0	

#### ANUNCIO\_IMG

Comentarios de la tabla: almacena los identificadores de las imágenes

Columna	Tipo	Nulo	Predeterminado	Comentarios
id	int(11)	No		almacena el identificador de la imagen
id_anuncio	int(11)	No		

#### ARTICULO

Columna	Tipo	Nulo	Predeterminado	Comentarios
id	int(11)	No		identifica el articulo
nombre	varchar(100)	No		nombre del articulo
descripcion	varchar(1000)	No		descripcion del articulo
precio	float(10,2)	No		precio del articulo
img	Varchar(100)	No		imagen del articulo
id_plataforma	int(11)	No		

### **ARTICULOS\_ANUNCIOS**

Comentarios de la tabla: almacena los artículos de un anuncio(packs)

Columna	Tipo	Nulo	Predeterminado	Comentarios
id	int(11)	No		identifica los productos del anuncio
id_anuncio	int(11)	No		
id_articulo	int(11)	No		

### **ARTICULOS\_ACCESORIOS**

Columna	Tipo	Nulo	Predeterminado	Comentarios
id	int(11)	No		
capacidad	varchar(200)	No		
color	varchar(200)	No		

### **ARTICULOS\_CONSOLA**

Columna	Tipo	Nulo	Predeterminado	Comentarios
id	int(11)	No		
tipo	varchar(200)	No		
modelo	varchar(200)	No		
capacidad	varchar(200)	No		
formato	varchar(200)	No		
color	varchar(200)	No		

### **ARTICULOS\_JUEGO**

Columna	Tipo	Nulo	Predeterminado	Comentarios
id	int(11)	No		
formato	varchar(200)	No		
idioma	varchar(200)	No		
genero	varchar(200)	No		
desarrollador	varchar(200)	No		
edad	varchar(200)	No		

### **CARACTERISTICAS\_ACCESORIO**

Columna	Tipo	Nulo	Predeterminado	Comentarios
id	int(11)	No		Características de una consola en concreto
id_accesorio	int(11)	No		
id_atributos_accesorio	int(11)	No		

### **CARACTERISTICAS\_JUEGO**

Columna	Tipo	Nulo	Predeterminado	Comentarios
id	int(11)	No		Características de una consola en concreto
id_consola	int(11)	No		
id_atributos_consola	int(11)	No		

### **COMENTARIO**

Columna	Tipo	Nulo	Predeterminado	Comentarios
id	int(11)	No		identifica el comentario
titulo	varchar(100)	Sí	NULL	titulo del comentario
descripcion	varchar(1000)	No		descripcion del comentario
fecha	timestamp	No	CURRENT_TIMESTAMP	fecha de creación del comentario
id_usuario	int(11)	No		
id_anuncio	int(11)	No		

### **CONSOLA**

Columna	Tipo	Nulo	Predeterminado	Comentarios
id	int(11)	No		

### **DIRECCION\_ENVIO**

Columna	Tipo	Nulo	Predeterminado	Comentarios
id	int(11)	No		identifica la dirección de envío
direccion	varchar(1000)	No		dirección de envío
id_usuario	int(11)	No		
id_localidad	int(11)	No		
titulo	varchar(200)	No		

### **JUEGO**

Columna	Tipo	Nulo	Predeterminado	Comentarios
id	int(11)	No		
id_consola	int(11)	No		

### **LOCALIDAD**

Columna	Tipo	Nulo	Predeterminado	Comentarios
id	int(11)	No		
id_provincia	int(11)	No		
nombre	varchar(200)	No		
cp	int(5)	No		

## PAIS

Columna	Tipo	Nulo	Predeterminado	Comentarios
id	int(11)	No		identifica al pais
nombre	varchar(1000)	No		

## PEDIDO

Columna	Tipo	Nulo	Predeterminado	Comentarios
id	int(11)	No		identificadores de pedidos
fecha	timestamp	No	CURRENT_TIMESTAMP	fecha en la que se realiza el pedido
id_usuario	int(11)	No		
id_producto	int(11)	No		

## PLATAFORMA

Columna	Tipo	Nulo	Predeterminado	Comentarios
id	int(11)	No		identifica la plataforma
nombre	varchar(100)	No		nombre de la plataforma

## PROVINCIA

Columna	Tipo	Nulo	Predeterminado	Comentarios
id	int(11)	No		identifica la provincia
nombre	varchar(100)	No		nombre de la provincia
id_pais	int(11)	No		

## REPORTAR\_ANUNCIO

Columna	Tipo	Nulo	Predeterminado	Comentarios
id	int(11)	No		identifica el reporte del anuncio
descripcion	varchar(1000)	No		descripcion del reporte del anuncio
revisado	varchar(1)	No	0	
id_usuario	int(11)	No		
id_anuncio	int(11)	No		

## REPORTAR\_PRECIO

Columna	Tipo	Nulo	Predeterminado	Comentarios
id	int(11)	No		identifica el reporte
precio	float(10,2)	No		precio que se reporta
revisado	varchar(1)	No	0	
id_usuario	int(11)	No		
id_articulo	int(11)	No		

## SEGUIMIENTO\_PRODUCTO

Columna	Tipo	Nulo	Predeterminado	Comentarios
id	int(11)	No		identifica el seguimiento del producto
id_usuario	int(11)	No		
id_producto	int(11)	No		

## USUARIO

Columna	Tipo	Nulo	Predeterminado	Comentarios
id	int(11)	No		identifica el usuario
nombre	varchar(100)	No		nombre del usuario
apellido	varchar(100)	No		apellidos del usuario
email	varchar(100)	No		email del usuario
user	varchar(100)	No		identificador login del usuario
password	varchar(100)	No		psw del usuario para login
fecha_nacimiento	date	No		fecha de nacimiento del usuario
telefono	int(11)	Sí	NULL	telefono del usuario
imagen	varchar(100)	Sí	NULL	identifica la imagen del usuario
baja	varchar(1)	No	0	0 usuario activo y 1 usuario

## *Anexo 2*

### Modelo Lógico

**ACCESORIO** (id)

**CP:** id

**ANUNCIO** (id, titulo, descripción, precio, id\_usuario)

**CP:** id

**C.Aje:** id\_usuario → USUARIO

**ANUNCIO\_IMG** (id, id\_anuncio)

**CP:** id

**C.Aje:** id\_anuncio → ANUNCIO

**ARTICULO** (id, nombre, descripción, precio, img, id\_plataforma)

**CP:** id

**C.Aje:** id\_plataforma → PLATAFORMA

**ATR\_CAPACIDAD** (id, nombre)

**CP:** id

**ATR\_COLOR** (id, nombre)

**CP:** id

**ATR\_DESARROLLADOR** (id, nombre)

**CP:** id

**ATR\_EDAD** (id, nombre)

**CP:** id

**ATR\_FORMATO** (id, nombre)

**CP:** id

**ATR\_GENERO** (id, nombre)

**CP:** id

**ATR\_IDIOMA** (id, nombre)

**CP:** id

**ATR\_MODELO** (id, nombre)

**CP:** id

**ATR\_TIPO** (id, nombre)

**CP:** id

**COMENTARIO** (id, titulo, descripción, fecha, id\_usuario, id\_anuncio)

**CP:** id

**C.Aje:** id\_usuario → USUARIO

**C.Aje:** id\_anuncio → ANUNCIO

**CONSOLA** (id)

**CP:** id

**DIRECCION\_ENVIO** (id, dirección, id\_usuario, id\_localidad, id\_pais, id\_provincia)

**CP:** id

**C.Aje:** id\_usuario → USUARIO

**C.Aje:** id\_localidad → LOCALIDAD

**C.Aje:** id\_pais → PAIS

**C.Aje:** id\_provincia → PROVINCIA

**JUEGO** (id)

**CP:** id

**LOCALIDAD** (id, nombre, cp, id\_provincia, id\_pais)

**CP:** id

**C.Aje:** id\_pais → PAIS

**C.Aje:** id\_provincia → PROVINCIA



**PAIS** (id, nombre)

**CP:** id

**PEDIDO** (id, fecha, id\_usuario, id\_producto)

**CP:** id

**C.Aje:** id\_usuario → USUARIO

**C.Aje:** id\_producto → PRODUCTO

**PLATAFORMA** (id, nombre)

**CP:** id

**PRODUCTO** (id, id\_anuncio, id\_articulo)

**CP:** id

**C.Aje:** id\_anuncio → ANUNCIO

**C.Aje:** id\_articulo → ARTICULO

**PROVINCIA** (id, nombre, id\_pais)

**CP:** id

**C.Aje:** id\_pais → PAIS

**REL\_CAPACIDAD** (id\_capacidad, id\_consola, id\_accesorio)

**CP:** (id\_capacidad, id\_consola, id\_accesorio)

**C.Aje:** id\_capacidad → ATR\_CAPACIDAD

**C.Aje:** id\_consola → CONSOLA

**C.Aje:** id\_accesorio → ACESORIO

**REL\_COLOR** (id\_color, id\_consola, id\_accesorio)

**CP:** (id\_color, id\_consola, id\_accesorio)

**C.Aje:** id\_color → ATR\_COLOR

**C.Aje:** id\_consola → CONSOLA

**C.Aje:** id\_accesorio → ACESORIO

**REL\_DESARROLLADOR** (id\_desarrollador, id\_juego)

**CP:** (id\_desarrollador, id\_juego)

**C.Aje:** id\_desarrollador → ATR\_DESARROLLADOR

**C.Aje:** id\_juego → JUEGO

**REL\_EDAD** (id\_edad, id\_juego)

**CP:** (id\_edad, id\_juego)

**C.Aje:** id\_edad → ATR\_EDAD

**C.Aje:** id\_juego → JUEGO

**REL\_FORMATO** (id\_formato, id\_consola, id\_juego)

**CP:** (id\_formato, id\_consola, id\_juego)

**C.Aje:** id\_formato → ATR\_FORMATO

**C.Aje:** id\_consola → CONSOLA

**C.Aje:** id\_juego → JUEGO

**REL\_GENERO** (id\_genero, id\_juego)

**CP:** (id\_genero, id\_juego)

**C.Aje:** id\_genero → ATR\_GENERO

**C.Aje:** id\_juego → JUEGO

**REL\_IDIOMA** (id\_idioma, id\_juego)

**CP:** (id\_idioma, id\_juego)

**C.Aje:** id\_idioma → ATR\_IDIOMA

**C.Aje:** id\_juego → JUEGO

**REL\_MODELO** (id\_modelo, id\_consola)

**CP:** (id\_modelo, id\_consola)

**C.Aje:** id\_modelo → ATR\_MODELO

**C.Aje:** id\_consola → CONSOLA

**REL\_TIPO** (id\_tipo, id\_consola)

**CP:** (id\_tipo, id\_consola)

**C.Aje:** id\_tipo → ATR\_TIPO

**C.Aje:** id\_consola → CONSOLA

**REPORTAR\_ANUNCIO** (id, descripción, revisado, id\_usuario, id\_anuncio)

**CP:** id

**C.Aje:** id\_usuario → USUARIO

**C.Aje:** id\_anuncio → ANUNCIO

**REPORTAR\_PRECIO** (id, precio, revisado, id\_usuario, id\_articulo)

**CP:** id

**C.Aje:** id\_usuario → USUARIO

**C.Aje:** id\_articulo → ARTICULO

**SEGUIMIENTO\_PRODUCTO** (id, id\_usuario, id\_producto)

**CP:** id

**C.Aje:** id\_usuario → USUARIO

**C.Aje:** id\_producto → PRODUCTO

**USUARIO** (id, nombre, apellido, email, user, password, fecha\_nacimiento, teléfono, baja)

**CP:** id

## Anexo 3

### Procedimientos

#### ▪ Sp\_DameUsuario

##### **Recibe:**

- nombre
- password

##### **Realiza:**

- Comprobación en la tabla “usuario”.

##### **Devuelve:**

- Si existe un usuario con dicho nombre y contraseña.

##### *Notas Adicionales:*

- El password estará cifrado.

#### ▪ Sp\_InsertaUsuario

##### **Recibe:**

- nombre
- apellido
- email
- nombre de usuario
- contraseña
- fecha de nacimiento
- teléfono
- imagen de perfil

##### **Realiza:**

- Inserción en la tabla “usuario”.

##### **Devuelve:**

- “**El nombre de usuario ya está en uso**”: cuando existe un usuario previamente registrado con el nombre de usuario recibido por parámetro.
- “**Registro realizado con éxito**”: cuando se realiza la inserción en la base de datos de forma correcta.

▪ Sp BorraUsuario

**Recibe:**

- código identificativo de un usuario (id\_usuario)

**Realiza:**

- Actualización en la tabla “usuario”

**Procedimiento:**

- Actualiza el usuario, asignándole al campo “baja” un 1. Los usuarios con un 1 en el campo baja, son usuarios no activos de la aplicación.

▪ Sp InsertaConsola

**Recibe:**

- nombre
- descripción
- precio
- imagen
- plataforma a la que pertenece la consola

**Realiza:**

- Inserción en la tabla “artículo” y en la tabla “consola”.

**Procedimiento:**

- Inserción de un artículo y en la tabla consola almacena el id de la consola insertada.

▪ Sp InsertaVideojuego

**Recibe:**

- nombre
- descripción
- precio
- imagen
- plataforma a la que pertenece el videojuego

**Realiza:**

- Inserción en la tabla “artículo” y en la tabla “juego”.

**Procedimiento:**

- Inserción de un artículo y en la tabla juego almacena el id del videojuego insertado.

- Sp InsertaAtributosConsola

**Recibe:**

- id\_consola
- tipo (puede tener los valores “sobremesa” o “portátil”)
- modelo (“FAT”, “SLIM”, etc.)
- capacidad (500Gb, etc.)
- formato (NTSC, PAL o NTSC-J)
- color

**Realiza:**

- Inserta en la tabla “atributos\_consola” y en la tabla “características\_consola”.

**Procedimiento:**

- Inserta los atributos de consola, el id consola e id atributo en “características\_consola”.

- SP DameTodasConsolas

**Realiza:**

- Consulta en la tabla “articulo”

**Devuelve:**

- Todas las consolas de la base de datos.

- Sp DameConsolasPorPlataforma

**Recibe:**

- id de plataforma

**Realiza:**

- Consulta en la tabla “articulo”

**Devuelve:**

- Todas las consolas que pertenezcan a dicha plataforma.

▪ Sp\_BuscarArticulo

**Realiza:**

- Consulta en la tabla “articulo”

**Devuelve:**

- Todos los artículos (consolas, videojuegos, accesorios) que tengan en el nombre la cadena a buscar recibida por parámetro.

▪ Sp\_ActualizarArticulo

**Recibe:**

- descripción
- precio
- imagen
- plataforma a la que pertenece el artículo.

**Realiza:**

- Actualización en la tabla “articulo”

**Procedimiento:**

- Actualiza los datos de un artículo ya sea una consola, videojuego o accesorio.

▪ Sp\_ActualizarUsuario

**Recibe:**

- Campos a modificar
- Identificador del usuario a modificar

**Realiza:**

- Actualización en la tabla “usuario”

**Procedimiento:**

- Permite actualizar los campos del usuario, todos salvo el nombre de usuario.

▪ Sp\_DameAtributosConsola

**Recibe:**

- código identificador de la consola.

**Realiza:**

- Consulta en la tabla “atributos\_consola”

**Devuelve:**

- Todos los atributos pertenecientes a la consola recibida por parámetros.

- Sp ActualizarAtributosJuego

**Recibe:**

- código identificador del atributo a modificar.

**Realiza:**

- Actualización en la tabla “atributos\_juego”

**Procedimiento:**

- Actualizar los atributos de un juego

- Sp DameTodosJuegos

**Devuelve:**

- Todos los videojuegos de la base de datos.

- Sp DameJuegosPorPlataforma

**Recibe:**

- código de plataforma.

**Realiza:**

- Consulta en la tabla “articulo”

**Devuelve:**

- Todos los videojuegos que pertenezcan a la plataforma recibida por parámetros.

- Sp DameJuegosPorConsola

**Recibe:**

- Código identificador de la consola.

**Realiza:**

- Consulta en la tabla “articulo”.

**Devuelve:**

- Todos los juegos que pertenezcan a la consola recibida por parámetro.

- Sp InsertaAccesorio

**Recibe:**

- nombre



- descripción
- precio
- imagen
- plataforma a la que pertenece el accesorio.

**Realiza:**

- Inserción en la tabla artículos y en la tabla accesorio

**Procedimiento:**

- Inserción en la tabla artículos y en la tabla accesorio almacena el id del accesorio que acabamos de insertar y el id de la consola a la que pertenece el accesorio.

▪ Sp\_DameAtributosJuego

**Recibe:**

- id de un juego

**Realiza:**

- Consulta en la tabla “atributos\_juego”.

**Devuelve:**

- Todos sus atributos.

▪ Sp\_InsertaAtributosAccesorio

**Recibe:**

- id de un accesorio

**Realiza:**

- Inserta en la tabla “atributos\_accesorio”.

**Procedimiento:**

- Inserta una serie de características sobre ese accesorio (capacidad y color)

▪ Sp\_InsertarAtributosJuego

**Recibe:**

- id del juego

**Realiza:**

- Consulta en la tabla “atributos\_juego”.

**Procedimiento:**

- Inserta una serie de características sobre ese juego (formato, idioma, género, desarrollador, edad)

- Sp ActualizarAtributoAccesorio

**Recibe:**

- El código identificador del atributo de accesorio a modificar.

**Realiza:**

- Actualiza en la tabla “atributos\_accesorio”.

**Procedimiento:**

- Actualizar los atributos de un accesorio pasado por parámetro.

- Sp DameTodosAccesorios

**Realiza:**

- Consulta en la tabla “articulo”.

**Devuelve:**

- Todos los accesorios.

- Sp DameAccesoriosConsola

**Recibe:**

- El código identificador de una consola.

**Realiza:**

- Consulta en la tabla “articulo”.

**Devuelve:**

- Todos los accesorios que pertenezcan a la consola pasada por parámetro.

- Sp DameAccesoriosPorPlataforma

**Recibe:**

- El código identificador de una plataforma.

**Realiza:**

- Consulta en la tabla “articulo”.

**Devuelve:**

- Todos los accesorios que pertenezcan a la plataforma recibida por parámetros.

▪ Sp InsertaComentario

**Recibe:**

- título
- descripción
- id\_usuario
- id\_anuncio

**Realiza:**

- Inserción en la tabla comentarios

**Procedimiento:**

- Permite insertar a cualquier usuario un comentario

*Notas Adicionales:*

- El id se autoincrementa por lo que no lo pasamos por parámetros
- La fecha no la pasamos por parámetro se pone por defecto.

*Mejoras:*

- Comprobar si el usuario a introducido más de 15 comentarios en menos de 5 min (para evitar spam).

▪ Sp BorrarComentario

**Recibe:**

- id del comentario a borrar.

**Realiza:**

- Borra un comentario

**Procedimiento:**

- Permite borrar al administrador los comentarios

*Notas Adicionales:*

- Al borrar un anuncio deberemos borrar todos los comentarios del mismo.

▪ Sp ActualizarComentario

**Recibe:**

- id
- título
- descripción.

**Realiza:**

- Editar un comentario

**Procedimiento:**

- Permite al administrador editar comentarios pasando

- Sp\_DameComentarioPorUsuario

**Recibe:**

- id\_usuario

**Devuelve:**

- Los comentarios de un usuario pasado por parámetro.

Sp\_DameComentarioPorUsuario

**Recibe:**

- id\_anuncio

**Devuelve:**

- Los comentarios de un anuncio pasado por parámetro.

- Sp\_InsertarDireccionEnvio

**Recibe:**

- dirección
- id\_usuario
- id\_localidad
- titulo

**Realiza:**

- Inserta una dirección de envío (“Registro realizado con éxito”)

**Procedimiento:**

- Permite insertar al usuario direcciones de envío.

*Comprobamos:*

- Si la dirección y localidad esta repetida (“La dirección esta repetida”)

- Sp\_BorrarDireccionEnvio

**Recibe:**

- id de la dirección

**Realiza:**

- Borra una dirección de envío

**Procedimiento:**

- Permite a los usuarios borrar una dirección de envío pasada por parámetro.

- Sp EditarDireccionEnvio

**Recibe:**

- id
- dirección
- id\_localidad
- id\_usuario
- título.

**Realiza:**

- Editamos la dirección de envío.

**Procedimiento:**

- Permite editar a los usuarios su dirección de envío pasada por parámetro.

*Comprobamos:*

- Si la dirección y localidad esta repetida (“La dirección esta repetida”)

- Sp DameDireccionesEnvioPorUsuario

**Recibe:**

- id\_usuario.

**Devuelve:**

- Las direcciones de envío de un usuario pasado por parámetro.

- Sp InsertarPlataforma

**Recibe:**

- Nombre de la plataforma

**Realiza:**

- Inserta una plataforma

**Procedimiento:**

- Permite insertar al administrador plataformas pasadas por parámetro.

▪ Sp\_BorrarPlataforma

**Recibe:**

- id de la plataforma

**Realiza:**

- Borra una plataforma

**Procedimiento:**

- Permite borrar al administrador plataformas pasadas por parámetro.

▪ Sp\_DamePaises

**Devuelve:**

- Todos los países

Sp\_DameProvinciaPorPais

**Recibe:**

- id\_pais

**Devuelve:**

- Todas las provincias de un país pasado por parámetro.

▪ Sp\_DameLocalidadPorProvincia

**Recibe:**

- id\_provincia

**Devuelve:**

- Las localidades de una provincia pasada por parámetro.

▪ Sp\_InsertarPedido

**Recibe:**

- id\_usuario
- id\_producto

**Realiza:**

- Inserta un pedido

**Procedimiento:**

- Permite al usuario insertar un pedido pasado por parámetro.

▪ Sp\_BorrarPedido

**Recibe:**

- id\_pedido

**Realiza:**

- Borra un pedido

**Procedimiento:**

- Permite al usuario borrar un pedido pasado por parámetro.

▪ Sp\_DamePedidosPorUsuario

**Recibe:**

- id\_usuario

**Devuelve:**

- Los pedidos comprados por un usuario pasado por parámetro.

▪ Sp\_DamePedidosPorProducto

**Recibe:**

- id\_producto

**Devuelve:**

- Los pedidos realizados para un producto pasado por parámetro.

▪ Sp\_DameAnuncios

**Devuelve:**

- Todos los anuncios

▪ Sp\_InsertaAnuncio

**Recibe:**

- titulo
- descripción
- precio
- id\_usuario

**Realiza:**

- Inserción en la tabla anuncio y en la tabla artículos\_anuncio

**Procedimiento:**

- Inserción en la tabla anuncio y en la tabla artículos\_anuncio almacena el id del anuncio que acabamos de insertar y el id del artículo que pertenece a ese anuncio.

- Sp ActualizarAnuncio

**Recibe:**

- descripción
- precio
- titulo
- id\_anuncio

**Realiza:**

- Actualización de la tabla anuncio

**Procedimiento:**

- Permite editar a los usuarios el anuncio pasada por parámetro.

- Sp DeshabilitarAnuncio

**Recibe:**

- Id\_anuncio

**Realiza:**

- Actualizamos la tabla “anuncio”

**Procedimiento:**

- Actualiza el anuncio pasado por parámetro, asignándole al campo “activo” un 1. Los anuncios con un 1 en el campo activo, son anuncios no habilitados de la aplicación los cuales se pondrá así si el artículo es comprado por alguien.

- Sp InsertaAnuncioImg

**Recibe:**

- Id\_anuncio

**Realiza:**

- Inserta en la tabla “anuncio\_img”

**Procedimiento:**

- Permite insertar al usuario imágenes a un anuncio pasado por parámetro.

*Comprobamos:*

- Si el anuncio existe sino (“Anuncio no existente”)



▪ Sp\_BorrarAnuncioImg

**Recibe:**

- Id\_anuncio\_img

**Realiza:**

- Borrar en la tabla “anuncio\_img”

**Procedimiento:**

- Permite borrar al usuario la imagen pasada por parámetro.

▪ Sp\_DameAnuncioImgs

**Recibe:**

- Id\_anuncio

**Realiza:**

- Consulta en la tabla “anuncio\_img”

**Devuelve:**

- Todas las imágenes de un anuncio pasado por parámetro.

▪ Sp\_InsertaReporteAnuncio

**Recibe:**

- descripción
- Id\_anuncio
- Id\_usuario

**Realiza:**

- Inserta en la tabla “reportar\_anuncio”

**Procedimiento:**

- Permite crear al usuario un reporte de un anuncio pasado por parámetro.

*Comprobamos:*

- Si el anuncio existe o es de ese usuario sino (“El anuncio no es de ese usuario o no existe”)

▪ Sp\_ActualizarReporteAnuncio

**Recibe:**

- descripción
- Id\_reporte

**Realiza:**

- Actualiza en la tabla “reportar\_anuncio”

**Procedimiento:**

- Permite actualizar al usuario un reporte de un anuncio pasado por parámetro.

▪ Sp\_ReporteAnuncioRevisado

**Recibe:**

- Id\_reporte\_anuncio

**Realiza:**

- Actualiza en la tabla “reportar\_anuncio”

**Procedimiento:**

- Actualiza un reporte de un anuncio pasado por parámetro asignándole al campo “revisado” un 1. Los anuncios reportados con un 1 en el campo revisado, son reportes de anuncio revisados de la aplicación.

▪ Sp\_DameReporteAnuncioNoRevisados

**Devuelve:**

- Los reportes de anuncios que no están revisados, es decir el campo revisado es “0”.

▪ Sp\_DameReporteAnuncioPorAnuncio

**Recibe:**

- id\_anuncio

**Devuelve:**

- Los reportes de anuncios realizados para un anuncio pasado por parámetro.

▪ Sp\_DameReporteAnuncioPorUsuario

**Recibe:**

- id\_usuario

**Devuelve:**

- Los reportes de anuncios realizados para un usuario pasado por parámetro.

▪ Sp DameReporteAnuncioRevisados

**Devuelve:**

- Los reportes de anuncios que están revisados, es decir el campo revisado es "1".

▪ Sp DameReporteAnuncioTodos

**Devuelve:**

- Los reportes de anuncios realizados por los usuarios.

▪ Sp InsertaReportePrecio

**Recibe:**

- descripción
- Id\_articulo
- Id\_usuario

**Realiza:**

- Inserta en la tabla "reportar\_precio"

**Procedimiento:**

- Permite crear al usuario un reporte de un precio de un artículo pasado por parámetro.

*Comprobamos:*

- Si el articulo existe ("El artículo no existe")

▪ Sp ActualizarReportarPrecio

**Recibe:**

- precio
- Id\_reporte

**Realiza:**

- Actualiza en la tabla "reportar\_precio"

**Procedimiento:**

- Permite actualizar al usuario un reporte de un precio pasado por parámetro.

▪ Sp\_ReportePrecioRevisado

**Recibe:**

- Id\_reporte\_precio

**Realiza:**

- Actualiza en la tabla “reportar\_precio”

**Procedimiento:**

- Actualiza un reporte de un precio pasado por parámetro asignándole al campo “revisado” un 1. Los precio reportados con un 1 en el campo revisado, son reportes de precio revisados de la aplicación.

▪ Sp\_DameReportePrecioNoRevisado

**Devuelve:**

- Los reportes de precios que no están revisados, es decir el campo revisado es “0”.

▪ Sp\_DameReportePrecioPorArticulo

**Recibe:**

- id\_articulo

**Devuelve:**

- Los reportes de precios realizados para un artículo pasado por parámetro.

▪ Sp\_DameReporteAnuncioPorUsuario

**Recibe:**

- id\_usuario

**Devuelve:**

- Los reportes de precios realizados para un usuario pasado por parámetro.

▪ Sp\_DameReportePrecioRevisados

**Devuelve:**

- Los reportes de precios que están revisados, es decir el campo revisado es “1”.

▪ Sp\_DameReportePrecioTodos

**Devuelve:**

- Los reportes de precios realizados por los usuarios.