[Título del documento]

[Subtítulo del documento]

Index

[OBJECTO DEL PROJECTO 1](#_Toc41674333)

[Definición del proyecto 1](#_Toc41674334)

[Tecnologías 1](#_Toc41674335)

[Requisitos mínimos 1](#_Toc41674336)

[Mínimos 2](#_Toc41674337)

[PLAN DE TRABAJO 3](#_Toc41674338)

[Plan de trabajo Inicial 3](#_Toc41674339)

[Plan de trabajo Real 4](#_Toc41674340)

[Contenido Técnico 5](#_Toc41674341)

[Diseño de interfaz 5](#_Toc41674342)

[Guía de estilos 5](#_Toc41674343)

[Consideraciones de Front End 4](#_Toc41674344)

[Interacción de la página con el usuario 4](#_Toc41674345)

[Formularios 5](#_Toc41674346)

[Consideraciones Back End 6](#_Toc41674347)

[Página de administración 6](#_Toc41674348)

[Bases de dades 7](#_Toc41674349)

[Disseny i eines 8](#_Toc41674350)

[Diagrama de casos d’ús 8](#_Toc41674351)

[Diagrama de classes 8](#_Toc41674352)

[Control de versions 8](#_Toc41674353)

[Programes utilizados 11](#_Toc41674354)

[Manual d’usuari del programa 0](#_Toc41674355)

[Pressupost 1](#_Toc41674356)

[Línies de millora 2](#_Toc41674357)

[Bibliografia o webgrafía 3](#_Toc41674358)

[ANNEX 4](#_Toc41674359)

[Buscar.js 4](#_Toc41674360)

# OBJECTO DEL PROJECTO

## Definición del proyecto

La intención de mi proyecto es desarrollar un sitio web que funcione como tienda de deportes.

En la página Web puede:

* Crear usuario
* Explorar productos
* Comprar productos
* Buscar/Filtrar productos
* Crear o quitar productos

## Tecnologías

Siendo un portal web, usaré **HTML, CSS3 y JavaScript,** usando bibliotecas como Bootstrap y **Jquery.** **AJAX** se utiliza, para trabajar de forma asincrónica, por ejemplo, para filtrar productos o para autenticar formularios.

Para poder interactuar con una base de **datos MYSQL,**, s’utilitzarà se utilizará **PHP.**.

La página Web se creará con Visual IDE **Studio Code**.

## Requisitos mínimos

Después de un estudio, puedo lograr los requisitos mínimos:

1. FrontEnd: El sitio web cumple con la apariencia de menús y formularios, y contenidos que se pueden actualizar de forma asincrónica.
2. BackEnd: El hecho de que la página tiene un inicio de sesión, podemos hacer que un usuario administrativo tenga la capacidad de modificar el contenido de la página.

En la base de datos, podemos almacenar los productos y precios de la tienda.

## Mínimos

|  |  |  |  |
| --- | --- | --- | --- |
| Nivel | Código | Apoyo | Profesor-Módulo |
| **Frontend** | F1 | Diseños de interfaz mediante HTML5/CSS3 que muestra contenido textual, formularios y multimedia. | Carlos M9 |
|  | F2 | Menús interactivos y validación desde el cliente. | Carlos M6 |
|  | F3 | Actualización asincrónica de contenido**(AJAX** Modificación del sol) | Carlos M6 |
|  | F4 | Autenticar usuarios con formulario. | Jordi M7 |
| **Backend** | B1 | Página de contenido administrativo (crear, leer, modificar y eliminar) | Jordi M7 |
|  | B2 | Seguridad y gestión de usuarios (alta, baja, modificación y eliminación), vinculando a los usuarios a diferentes perfiles. | Jordi M7 |
|  | B3 | Gestión de sesiones. | Jordi M7 |
| **-** | B4 | Gestión y movimientos históricos. | Jordi M7 |
|  | B5 | Uso de servicios web. | Jordi M7 |
| **Base Datos** | D1 | Creación de la base de datos | Jordi M2 |
|  | D2 | Optimización de consultas y uso de índices | Jordi M2 |
| **-** | D3 | Gestión histórica y movimientos mediante desencadenantes y procedimientos | Jordi M2 |
| **Diseño**  **(UML)** | U1 | Diagrama de caso de uso | Anna M5 |
|  | U2 | Diagrama de clases. | Jordi M7  Anna M5  Juan M3 |
| **Herramientas** | E1 | Control de versiones. | Anna M8 |

# PLAN DE TRABAJO

## Plan de trabajo Inicial

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Leyenda** |  | **Miércoles 13** | **Jueves 14** | **Viernes 15** | **Sábado 16** | **Domingo 17** |
| Frontend |  | 5h | 5h | 5h | 1h | 1h |
| Backend |  |  |  |  | 3h | 4h |
| Base de datos |  |  |  |  | 1h |  |
| Diagramas |  |  |  |  |  |  |
| Memoria |  |  |  |  |  |  |
| Lunes **18** | Martes **19** | Miércoles **20** | Jueves **21** | Viernes **22** | Sábado **23** | Domingo **24** |
| 1h | 1h | 1h | 1h | 1h | 1h | 1h |
| 4h | 4h | 4h | 4h | 4h | 4h | 4h |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| **Lunes 25** | **Martes 26** | **Miércoles 27** | **Jueves 28** | **Viernes 29** | **Sábado 30** | **Domingo 31** |
| 1h | 3h | 2h | 1h | 1h | 1h | 1h |
| 4h | 3h | 1h |  |  |  | 1h |
|  |  | 2h |  |  |  | 1h |
|  |  |  |  |  | 1h | 1h |
|  |  |  |  | 4h | 4h | 1h |
| Total | |  |  |  |  |  |
| Frontend | 34 h |  |  |  |  |  |
| Backend | 43 h |  |  |  |  |  |
| Base de datos | 4 h |  |  |  |  |  |
| Diagramas | 2 h |  |  |  |  |  |
| Memoria | 9 h |  |  |  |  |  |
| Horas Totales | 49 h |  |  |  |  |  |

## Plan de trabajo Real

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Leyenda** |  | **Miércoles 13** | **Jueves 14** | **Viernes 15** | **Sábado 16** | **Domingo 17** |
| Frontend |  | 6h | 6h | 3h |  |  |
| Backend |  |  |  | 3h | 4h | 5h |
| Base de datos |  |  |  |  | 2h |  |
| Diagramas |  |  |  |  |  | 1h |
| Memoria |  |  |  |  |  |  |
| **Lunes 18** | **Martes 19** | **Miércoles 20** | **Jueves 21** | **Viernes 22** | **Sábado 23** | **Domingo 24** |
| 2h | 1h | 1h | 1h | 1h | 1h | 1h |
| 4h | 5h | 5h | 5h | 5h | 5h | 5h |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| **Lunes 25** | **Martes 26** | **Miércoles 27** | **Jueves 28** | **Viernes 29** | **Sábado 30** | **Domingo 31** |
| 1h | 3h | 2h | 1h | 1h | 1h | 1h |
| 6h | 3h |  |  |  |  | 1h |
|  |  | 2h |  |  |  | 1h |
|  |  |  | 1h |  |  | 1h |
|  |  |  | 4h | 4h | 1h | 1h |
| Total | |  |  |  |  |  |
| Frontend | 33h |  |  |  |  |  |
| Backend | 55h |  |  |  |  |  |
| Base de datos | 4h |  |  |  |  |  |
| Diagramas | 2h |  |  |  |  |  |
| Memoria | 4h |  |  |  |  |  |
| Horas Totales | 98h |  |  |  |  |  |

# Contenido Técnico

## Diseño de interfaz

### Guía de estilos

***Tipografías***













***Paleta de colores***

**HEX:** #1E4B94

**RGB:** 30, 75, 148

**HEX:** #343A40

**RGB:** 52, 58, 64

**HEX:** #660204

**RGB:** 102, 2, 4

**HEX:** #F1F1F1

**RGB:** 230, 227, 227

***Iconografía***



***Imágenes***



Imagen índice (12 columnas)

1110 x 445.94



Imágenes deportes (3 columnas)

Altura 300



Imágenes deportes (12 columnas)

Altura 300px

Imagen Producto catalogo (3 columnas)

Altura 337px

Imagen Producto catalogo (12 columnas)

Altura 337px



Imagen producto (6 columnas)

Imagen Producto catalogo (12 columnas)

Altura 337px

Altura 414px

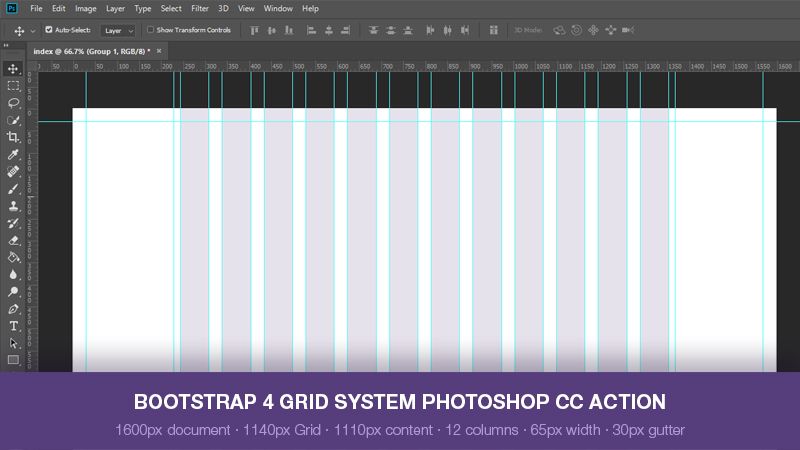


Imagen producto administración

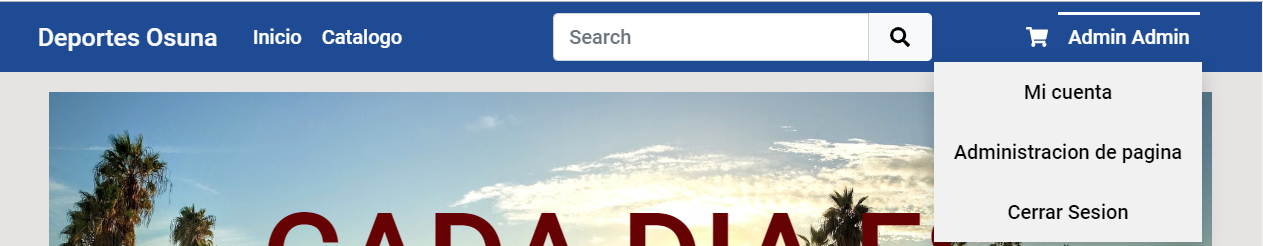
Tamaño 150px x 200px

***“Grid”***

Se utiliza el grid predeterminado de bootstrap.



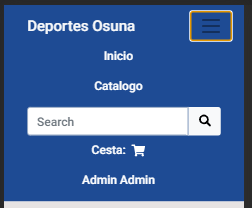
***Menús***



Roboto 16 Regular #FFFFFF

Altura 56px Regular

Fondo: #1E4B94



Roboto 16 Regular #FFFFFF

Altura 56px Regular

Fondo: #1E4B94

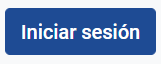
***Formularios***



Cajas de texto

Altura 30px

Roboto 16px Regular



Botones

Altura 38px

Fondo #1E4B94

Letra #FFFFFF



Botón eliminar

Altura 38px

Fondo #660204

Letra #212529



Botones

Altura 38px

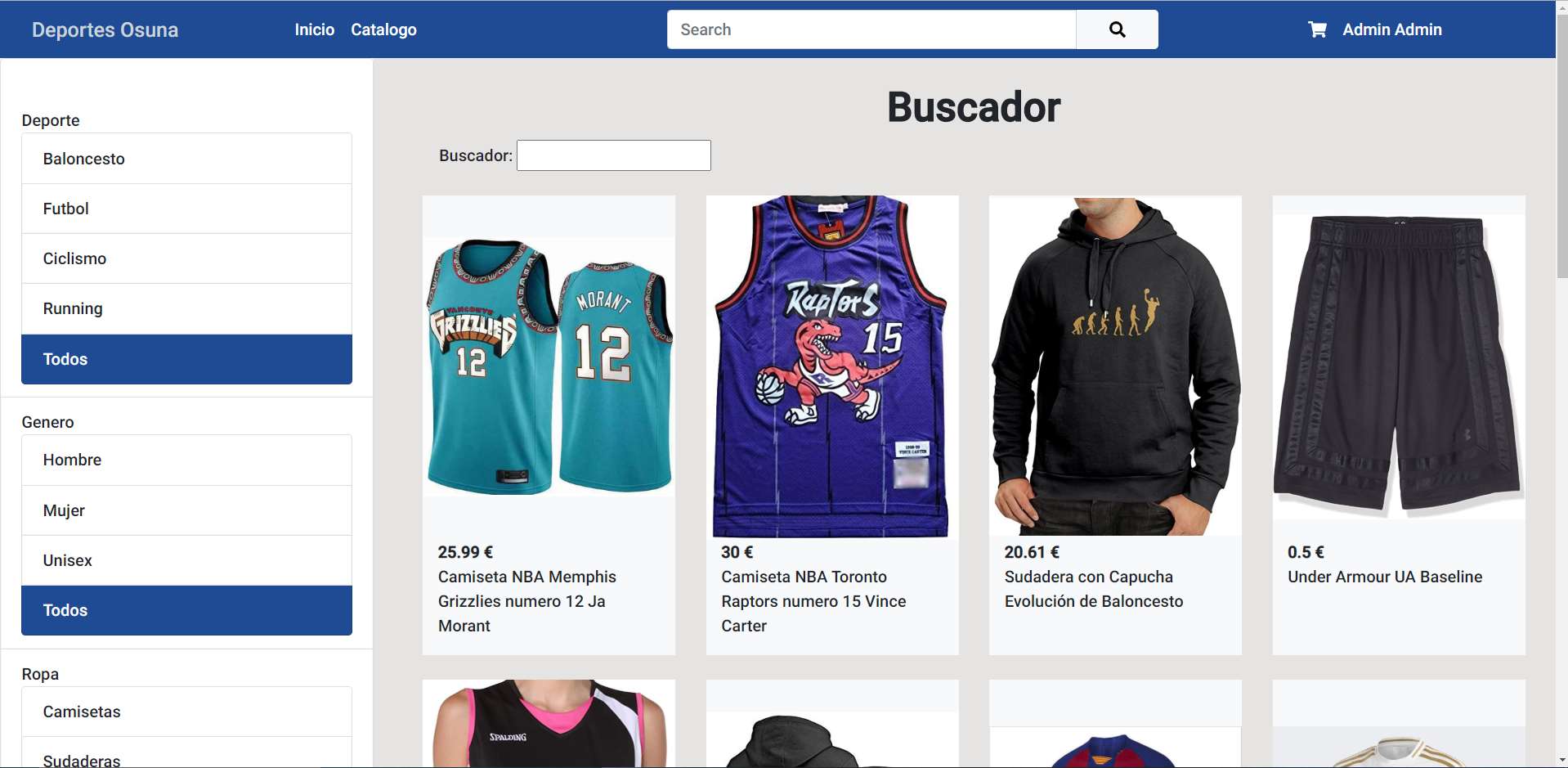
Fondo #660204

Letra #FFFFFF

## Consideraciones de Front End

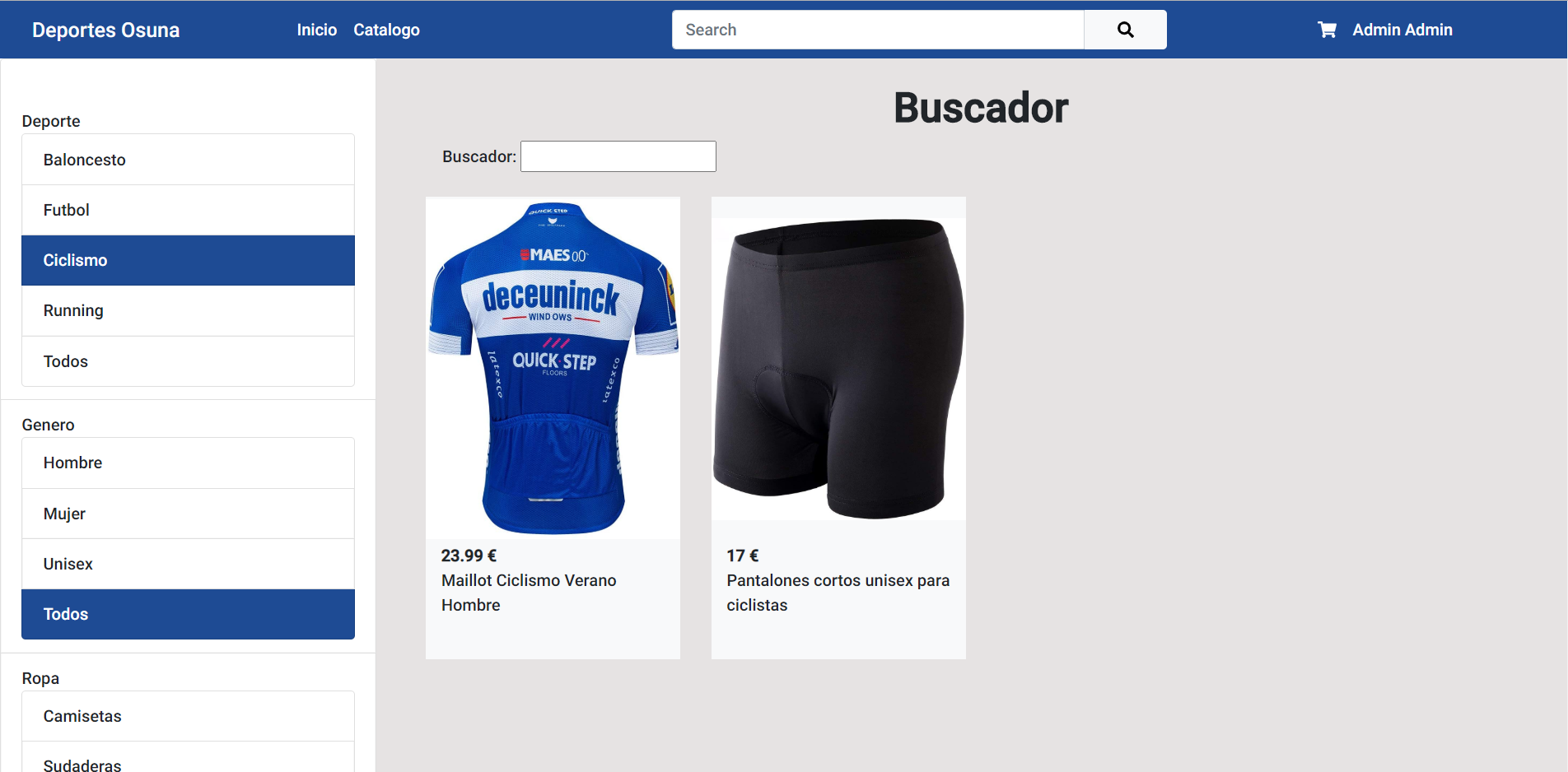
Este proyecto se ha realizado utilizando ***HTML5***, que se utiliza para hacer la base de la página, ***CSS3***, para añadirle estilos al código ***HTML***, ***JavaScript***, y las librerías de ***Bootstrap***, ***JQuery***, y ***Ajax***. A continuación, se cogerá la página **buscar.html** del proyecto para mostrar cómo se utilizan los elementos anteriormente mencionados.

### Interacción de la página con el usuario



Al cargar la página, podremos ver su diseño, que ha sido creado con HTML, CSS y Bootstrap.

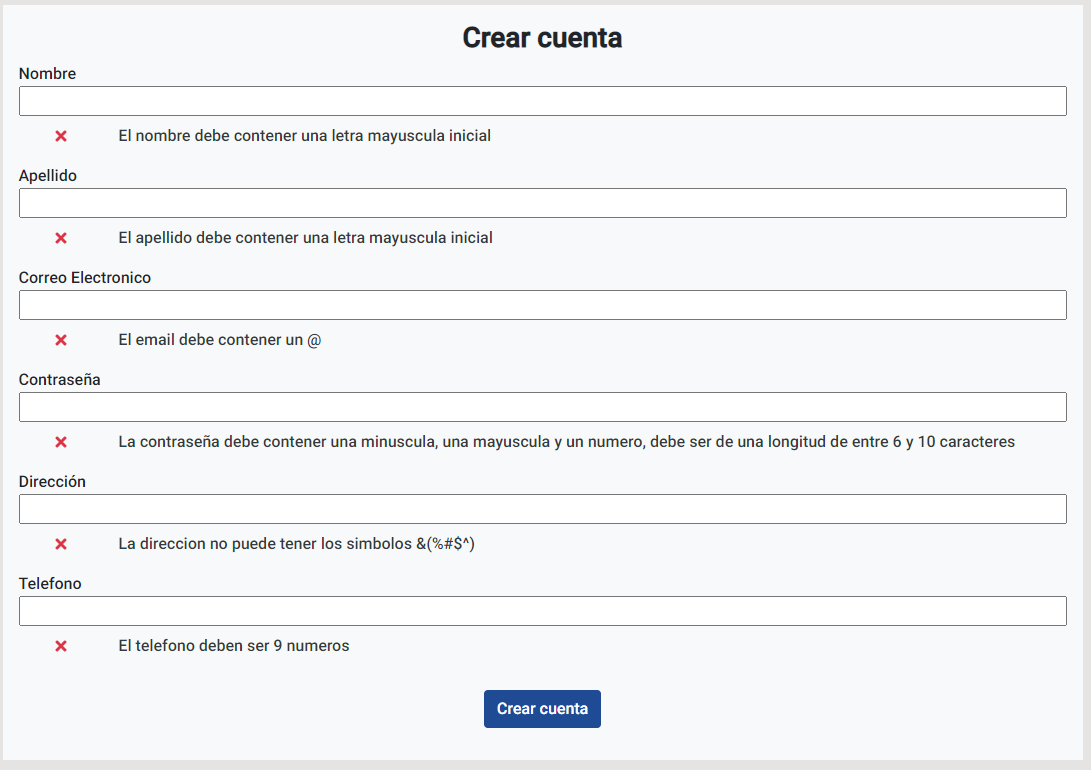
En el JavaScript, se ha creado un evento de JQuery, al hacer clic en una categoría, a la izquierda, llamara a un fichero PHP utilizando Ajax que rellenara la pagina web con los datos correspondientes.



### Formularios

Los campos se validan cada vez que pierden foco. Esto se realiza utilizando un evento de JQuery, al activarse se mira que el valor de ese campo concuerde con los valores que queremos meter en la base de datos, para ello se utiliza un “***Pattern***”.

**Crear usuario**



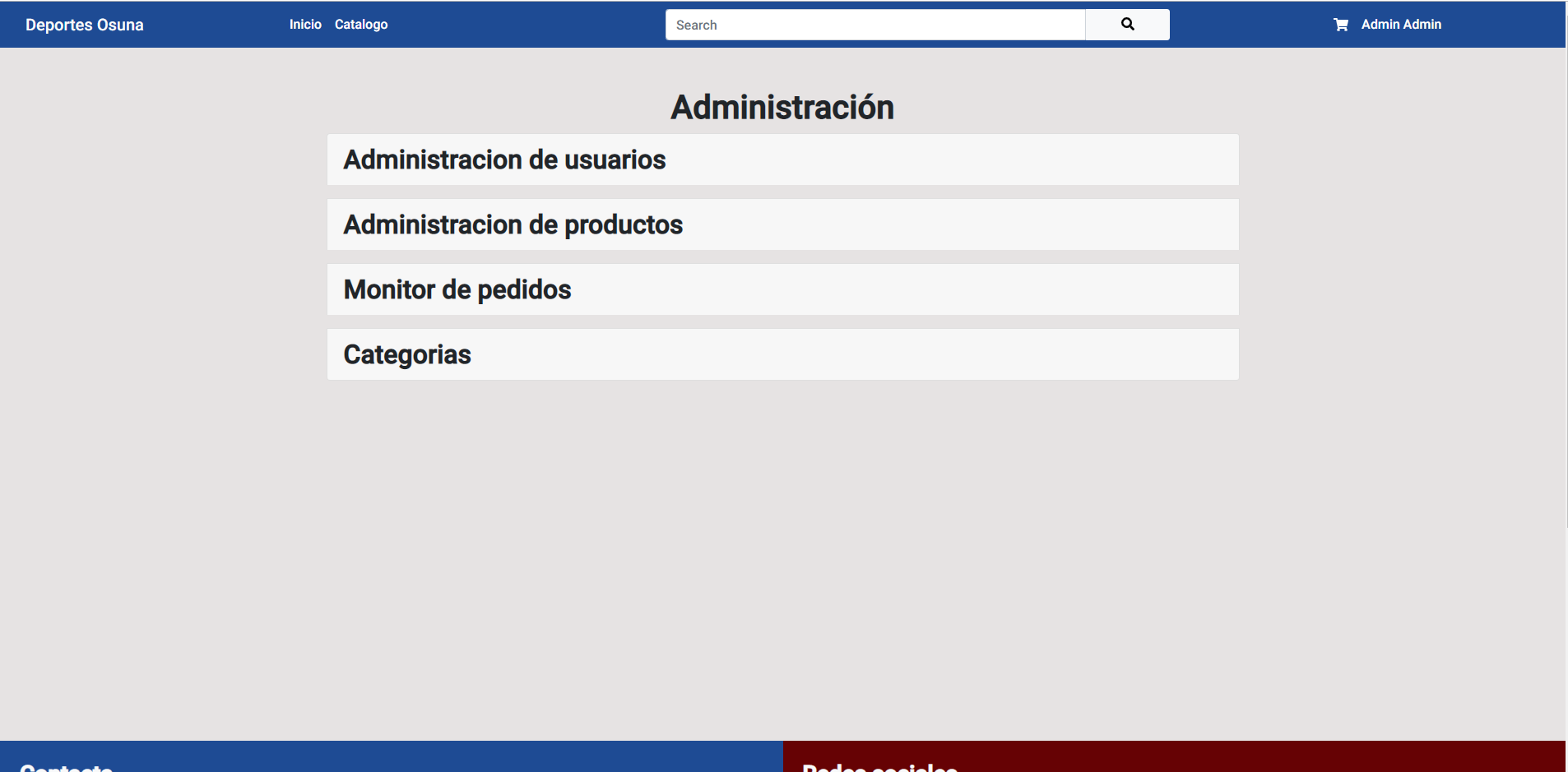
**Inicio de sesion**



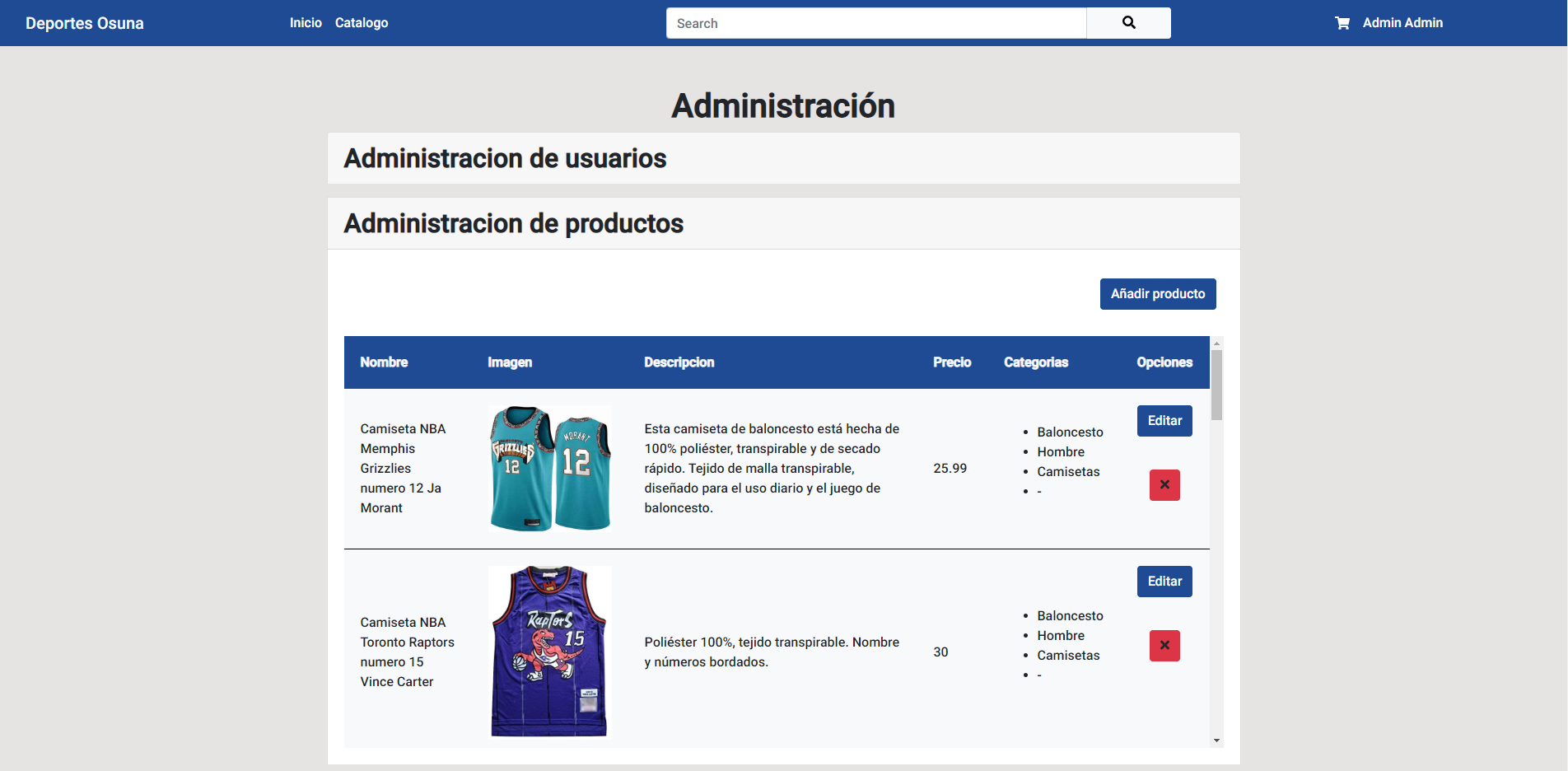
## Consideraciones Back End

### Página de administración

La pagina de administración esta dividida en 4 partes, usuarios, productos, pedidos y categorías.



Cada apartado tiene las herramientas para administrar necesarias para poder añadir, eliminar, y si el apartado lo permite, editar.

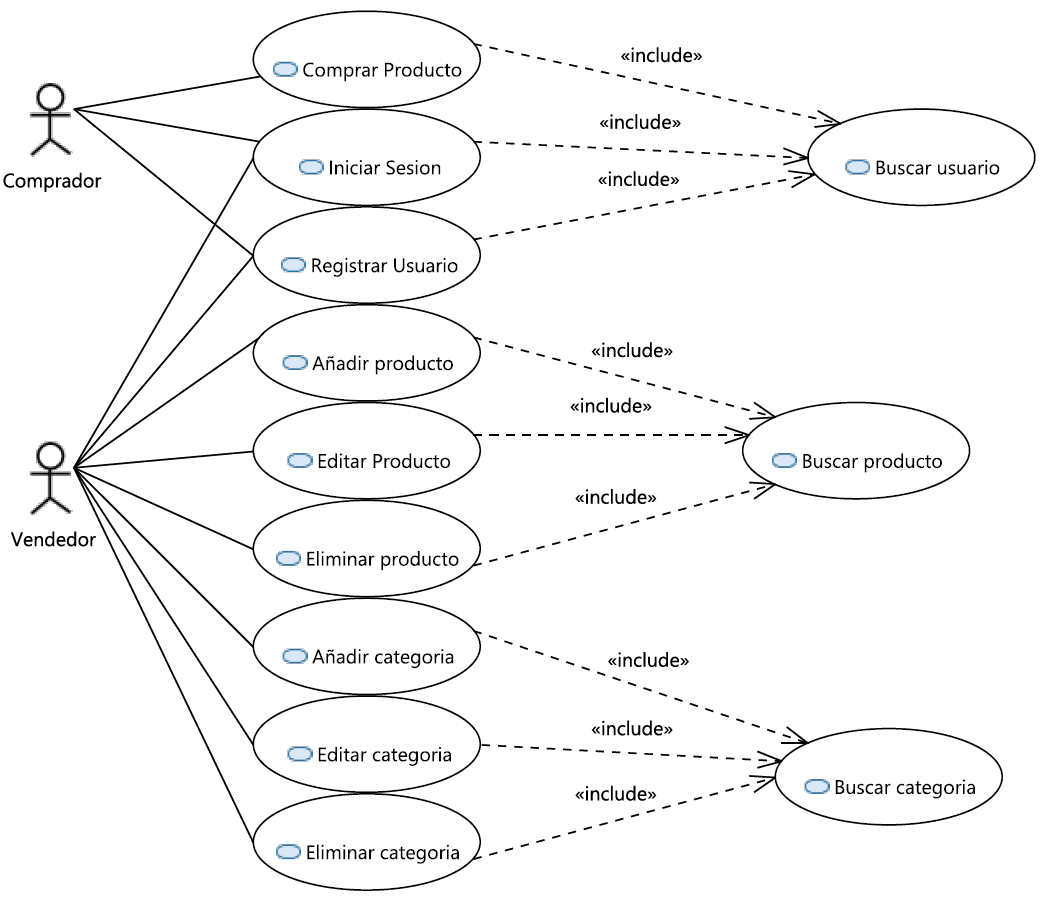


### seguretat i gestió d’usuari

## Bases de dades

## Disseny i eines

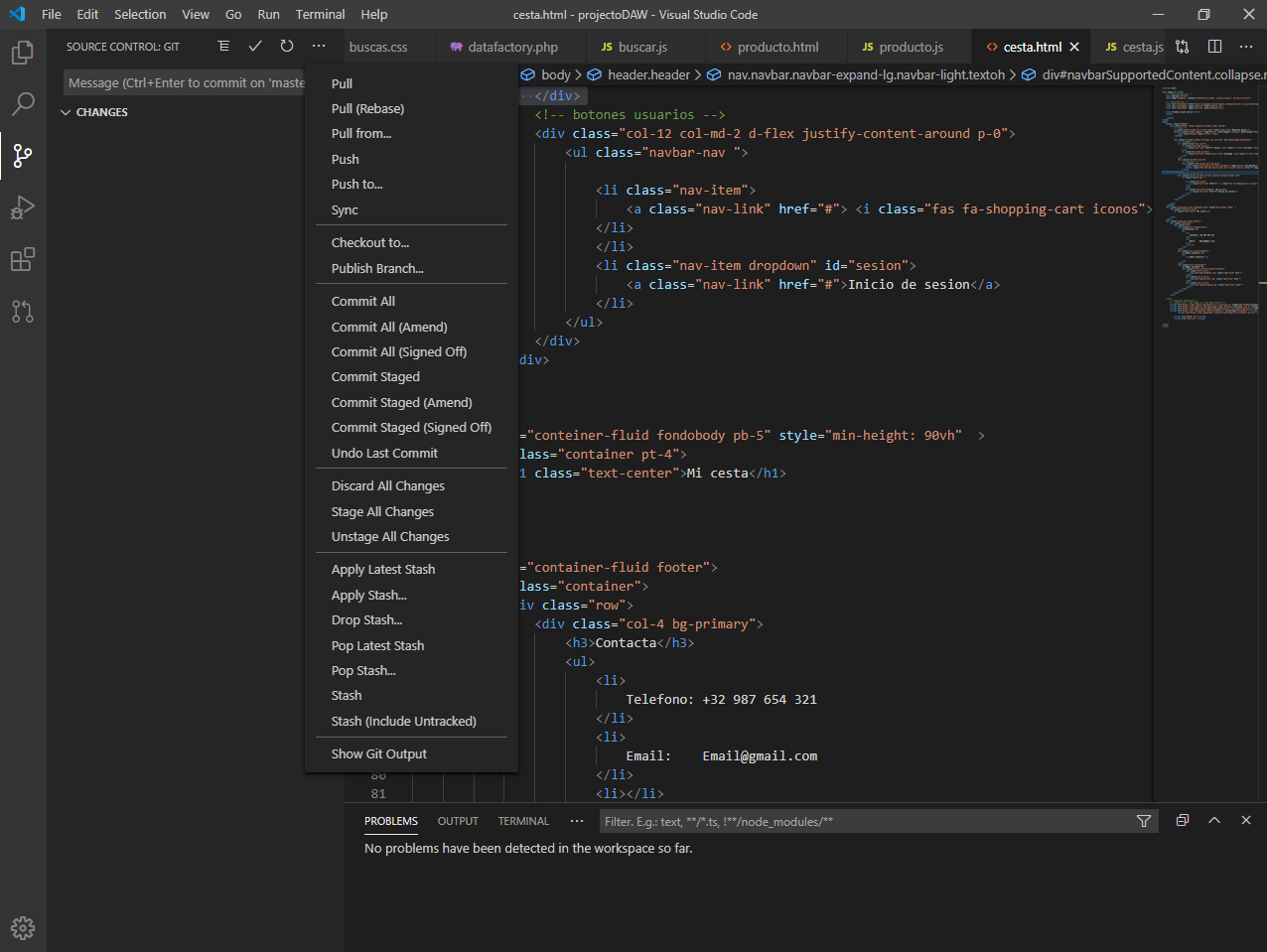
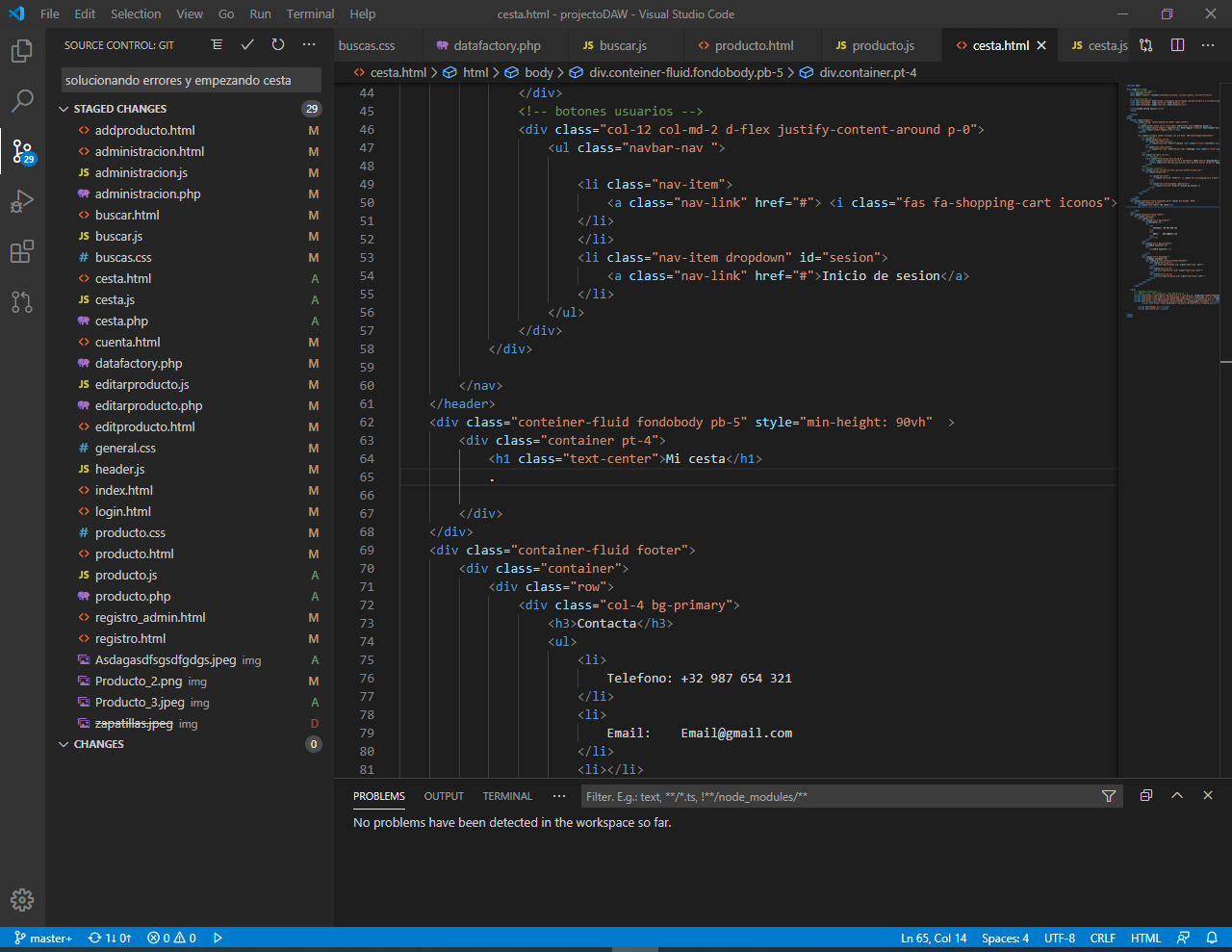
### Diagrama de casos d’ús



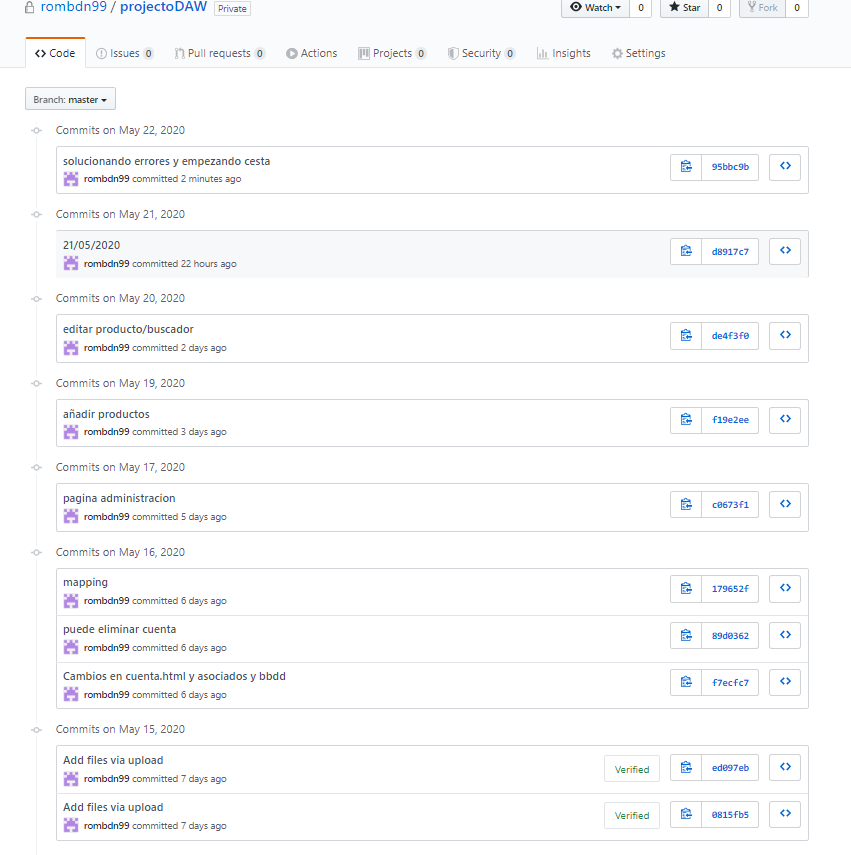
### Diagrama de classes

### Control de versions

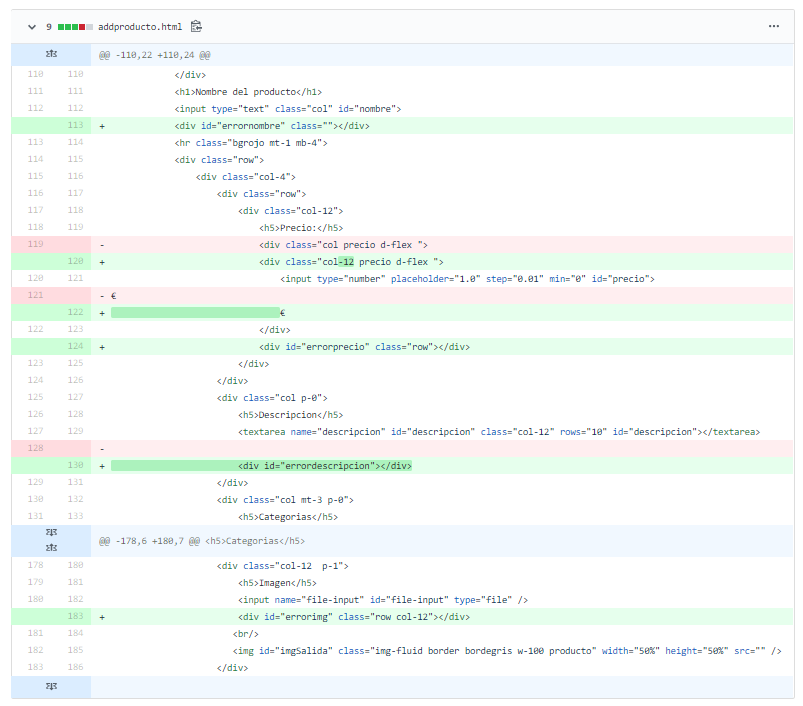
Hago el control de versiones con Github y Visual Studio Code. En visual estudio hay una pestaña donde te permite ver todos los archivos editados y hacer commit y push de esos ficheros.



***Commits primera semana***



***Comparación de diferentes documentos en 2 commits diferentes***





### Programes utilizados

A continuación, se muestra una serie de programas que se han utilizado para crear la página web.

**WAMP**

**Visual Studio Code**

Papyrus

Google Chrome

# Manual d’usuari del programa

# Pressupost

# Línies de millora

# Bibliografia o webgrafía

<https://www.w3schools.com/>

<https://www.php.net/manual/es/>

<https://stackoverflow.com/>

[https://getbootstrap.com/](https://getbootstrap.com/docs/4.5/getting-started/introduction/)

<https://developer.mozilla.org/es/docs/Web/JavaScript>

<http://www.planetb.ca/syntax-highlight-word>

<https://fontawesome.com/>

<https://www.flaticon.es/>

# ANNEX

## Buscar.js

1. $.post('buscar.php',{query: "getcategorias"})
2. .done(**function**(data,textStatus,jqXHR){
3. //console.log("Solicitud se ha completado correctamente "+ textStatus);
4. console.log(data);
5. **if**(data!="bad"){
6. $(".categorias").html(data)
7. }**else**{
8. console.log("No hay usuarios")
9. }
10. // location.replace("login.html")
11. eventos();
12. inicial();
13. })
14. .fail(**function**(data,textStatus,jqXHR){
15. console.log("Error: " + textStatus + ": " + jqXHR);
16. })
17. .always(**function**(data,textStatus,jqXHR){
18. console.log("buscar.php")
19. })
21. **function** inicial(){
22. selectini="select \* from productos"
23. **if**(**typeof** getParams(window.location.href)['deporte']!=='undefined'){
24. console.log("entra")
25. $("#d-"+getParams(window.location.href)['deporte']).addClass("activo");
26. buscador();
27. }**else** **if**(**typeof** getParams(window.location.href)['buscar']==='undefined'){
28. $.post('buscar.php',{query: "filtro",select: selectini})
29. .done(**function**(data,textStatus,jqXHR){
30. console.log("Solicitud se ha completado correctamente "+ textStatus);
31. ///console.log(data);
32. **if**(data!="bad"){
33. $("#buscador").html(data)
34. }**else**{
35. $("#buscador").html("<h1>No hay resultados</h1>")
36. }
37. // location.replace("login.html")
38. eventosproductos();
39. })
40. .fail(**function**(data,textStatus,jqXHR){
41. console.log("Error: " + textStatus + ": " + jqXHR);
42. })
43. .always(**function**(data,textStatus,jqXHR){
44. console.log("header.php")
45. })
46. }**else**{
47. console.log()
48. selectini+=" where nombre like ('%"+getParams(window.location.href)['buscar']+"%')";
49. $.post('buscar.php',{query: "filtro", select: selectini})
50. .done(**function**(data,textStatus,jqXHR){
51. console.log("Solicitud se ha completado correctamente "+ textStatus);
52. ///console.log(data);
53. **if**(data!="bad"){
54. $("#buscador").html(data)
55. }**else**{
56. $("#buscador").html("<h1>No hay resultados</h1>")
57. }
58. // location.replace("login.html")
59. eventosproductos();
61. })
62. .fail(**function**(data,textStatus,jqXHR){
63. console.log("Error: " + textStatus + ": " + jqXHR);
64. })
65. .always(**function**(data,textStatus,jqXHR){
66. console.log("header.php")
67. })
68. }
69. }
71. **function** eventos(){
73. $(".categoria").click(**function**(){
74. **if**($(**this**).hasClass("activo")){
75. $(**this**).removeClass("activo");
77. }**else**{
78. $(**this**).addClass("activo");
80. }
81. buscador();
82. })
83. $("#buscadorb").keyup(**function**(){
84. history.replaceState(**null**, '', '?buscar='+$(**this**).val());
85. buscador();
86. console.log($(**this**).val())
87. })
88. }
89. **function** eventosproductos(){
90. $(".productobody").click(**function**(){
91. console.log($(**this**).attr("id").replace('buscar',''));
92. location.replace("producto.html?id="+$(**this**).attr("id").replace('buscar',''))
93. })
94. }
96. **function** getParams (url) {
97. **var** params = {};
98. **var** parser = document.createElement('a');
99. parser.href = url;
100. **var** query = parser.search.substring(1);
101. **var** vars = query.split('&');
102. **for** (**var** i = 0; i < vars.length; i++) {
103. **var** pair = vars[i].split('=');
104. params[pair[0]] = decodeURIComponent(pair[1]);
105. }
106. **return** params;
107. };
108. console.log(getParams(window.location.href)['buscar'])
110. **function** buscador(){

113. select="select \* from productos ";
115. deporte=[];
116. d=0;
117. ropa=[];
118. r=0;
119. equip=[];
120. e=0;
121. genero=[];
122. g=0;
123. $('.activo').each(**function** () {
125. **switch**(**this**.id.split("-")[0]){
126. **case** "d":
127. **if**(**this**.id.split("-")[1]!=0){
129. deporte[d]=" deporte="+**this**.id.split("-")[1]
130. d++;
131. }

134. **break**;
135. **case** "r":
136. **if**(**this**.id.split("-")[1]!=0){
137. ropa[r]=" ropa="+**this**.id.split("-")[1]
138. r++
139. }
140. **break**;
141. **case** "e":
142. **if**(**this**.id.split("-")[1]!=0){
144. equip[e]=" equipamiento="+**this**.id.split("-")[1]
145. e++;
146. }
147. **break**;
148. **case** "g":
149. **if**(**this**.id.split("-")[1]!=0){
150. genero[g]=" genero="+**this**.id.split("-")[1]
151. g++;
152. }
153. **break**;
154. }
155. });
156. **if**(deporte.length>0 || ropa.length>0 || genero.length>0 || equip.length>0 || **typeof** getParams(window.location.href)['buscar']!=='undefined'){
157. where=" where";
158. }
159. filtros="";
160. **for**(**var** i=0;i<deporte.length;i++){
161. **if**(i==deporte.length-1){
162. filtros+=deporte[i]
164. }**else**{
165. filtros+=deporte[i]+" or";
167. }
169. }
170. **if**(deporte.length>0 && ropa.length>0){
171. filtros+=" and";
173. }
174. **for**(**var** i=0;i<ropa.length;i++){
175. **if**(i==ropa.length-1){
176. filtros+=ropa[i]
178. }**else**{
179. filtros+=ropa[i]+" or"
181. }
182. }
183. **if**((deporte.length>0 || ropa.length>0) && genero.length>0){
184. filtros+=" and";
185. }
186. **for**(**var** i=0;i<genero.length;i++){
187. **if**(i==genero.length-1){
188. filtros+=genero[i]
190. }**else**{
191. filtros+=genero[i]+" or"
193. }
194. }
195. **if**((deporte.length>0 || ropa.length>0 || genero.length>0) && equip.length>0){
196. filtros+=" and";
197. }
198. **for**(**var** i=0;i<equip.length;i++){
199. **if**(i==equip.length-1){
200. filtros+=equip[i]
202. }**else**{
203. filtros+=equip[i]+" or"
205. }
206. }
208. **if**(**typeof** getParams(window.location.href)['buscar']!=='undefined'){
209. **if**(deporte.length>0 || ropa.length>0 || genero.length>0 || equip.length>0){
210. filtros+=" and"
211. }
212. filtros+=" nombre like ('%"+getParams(window.location.href)['buscar']+"%')"
213. }
215. **if**(filtros!=""){
216. select+=where+filtros;
217. }
218. console.log(select)
219. $.post('buscar.php',{query: "filtro",select: select})
220. .done(**function**(data,textStatus,jqXHR){
221. console.log("Solicitud se ha completado correctamente "+ textStatus);
222. ///console.log(data);
223. **if**(data!="bad"){
224. $("#buscador").html(data)
225. }**else**{
226. $("#buscador").html("<h1>No hay resultados</h1>")
227. }
228. // location.replace("login.html")
229. //eventos();
231. })
232. .fail(**function**(data,textStatus,jqXHR){
233. console.log("Error: " + textStatus + ": " + jqXHR);
234. })
235. .always(**function**(data,textStatus,jqXHR){
236. console.log("filtro")
237. })
238. }