



# VIZUAL CONTENDER

Fernando Mosteiro Del Pilar  
fernando.mosteiro.pilardel@rai.usc.es

Desarrollo de Aplicaciones Web  
Septiembre 2019

## 1 Introducción

En este documento se profundiza sobre los primeros pasos en cuanto al desarrollo del sitio web de una empresa ficticia llamada **Vizual Contender**; la cual ofrece ayuda y asesoramiento en cuanto al modelo y diseño de prototipos de todo tipo de productos.

La idea de la empresa es que esta posea una gran variedad de herramientas y maquinaria altamente específica, que ayude de la misma manera a empresas y particulares de cualquier ámbito, que necesiten una ayuda a la hora de materializar sus ideas y obtener un producto final.

La página web será de gran ayuda para la empresa dado que expondrá a sus potenciales clientes de lo que es capaz, a través de un catálogo con sus trabajos más interesantes y que supusieran un desafío técnico. Además se dispondrá de un apartado de consulta en el que los clientes puedan tener sus primeros contactos con la empresa, recibiendo asesoría a partir de una explicación de la idea general siendo apoyada adjuntando archivos audiovisuales.

## 2 Inventario de Contenido

La empresa se reúne y se pone de acuerdo en lo que le interesa que aparezca en la página, exponiendo sus trabajos, premios y proveedores; sus capacidades técnicas y facilitar los primeros contactos con potenciales clientes.



Figure 1: Inventario de Contenido.

### 3 Arquitectura de la Información

La página se estructura de la siguiente manera:

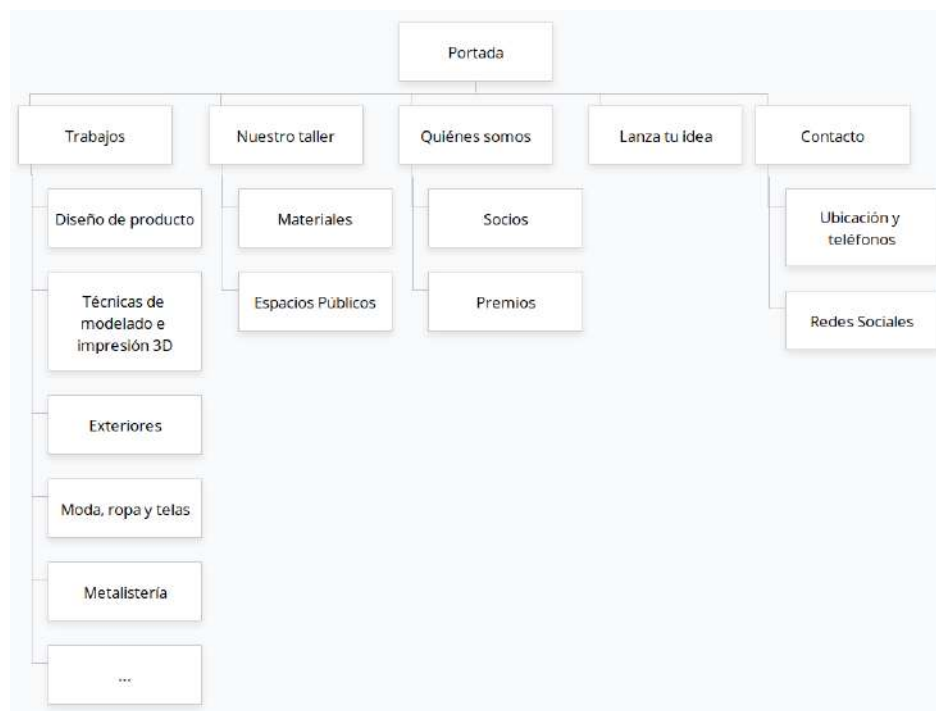


Figure 2: Arquitectura de la Información.

En el catálogo de *Trabajos* habrá una página para cada tipo; el apartado *Nuestro taller* puede concentrarse en una sola página, exponiendo las capacidades dentro y fuera del taller de la empresa; en *Quiénes somos* se presenta parte del personal más destacado y de renombre, junto con los socios y proveedores que ayudan fuera de las capacidades de la empresa; la página de *Lanza tu idea* se centra en las aportaciones de nuevos clientes además del feedback de los consultores; por último en *Contacto* se muestran demás datos de importancia.

## 4 Sketching

En la portada se puede ver el logo de la empresa, de fondo pequeños clips de unos pocos segundos con imágenes de calidad y ritmo de distintos momentos pertenecientes a momentos de trabajo en la empresa; similar al de la página de fotógrafo Dave Meyers.

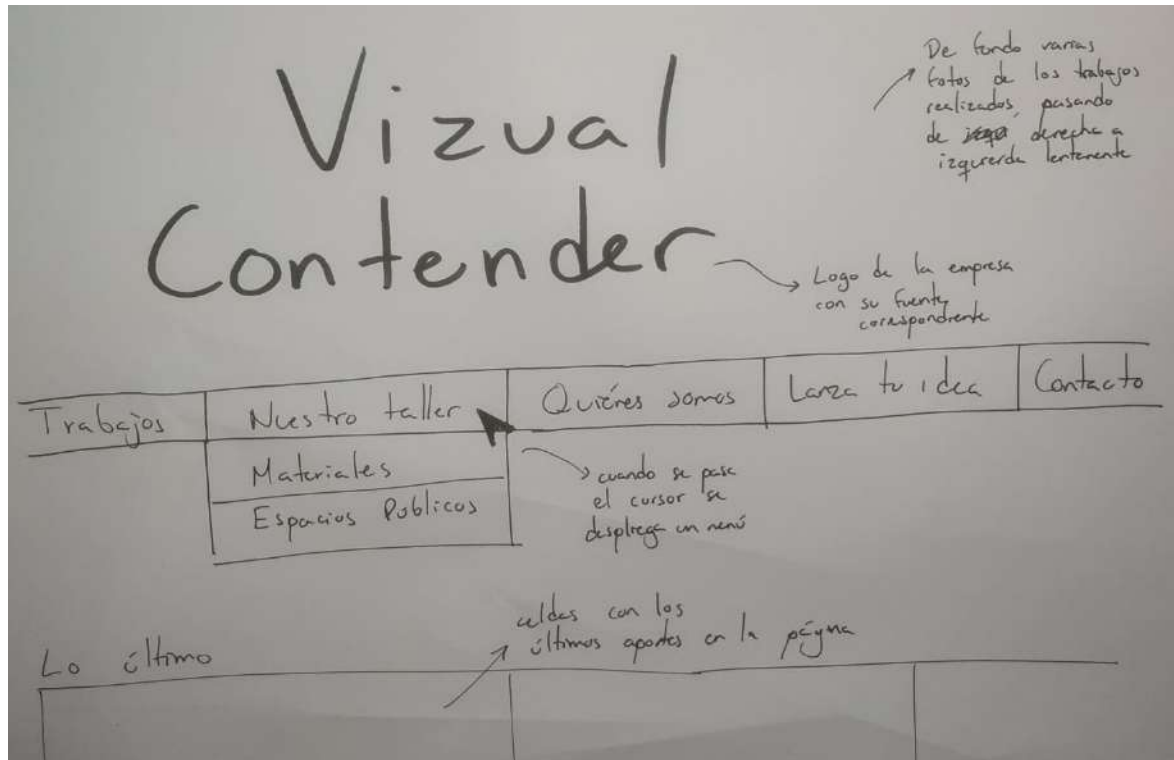


Figure 3: Portada.

Clicando en Trabajos, se pasa a una página intermedia con los principales o mejor valorados trabajos, más abajo unas celdas con fotografías dividiendo los trabajos en varias categorías según el tipo, similar a la página web del artista Felipe Pantone.

Para cada trabajo se relata el proceso desde los primeros inputs del cliente hasta el producto final, explicando los desafíos técnicos y como la empresa posee los medios necesarios para solucionarlos, estructurado de manera similar a la página de Nike.

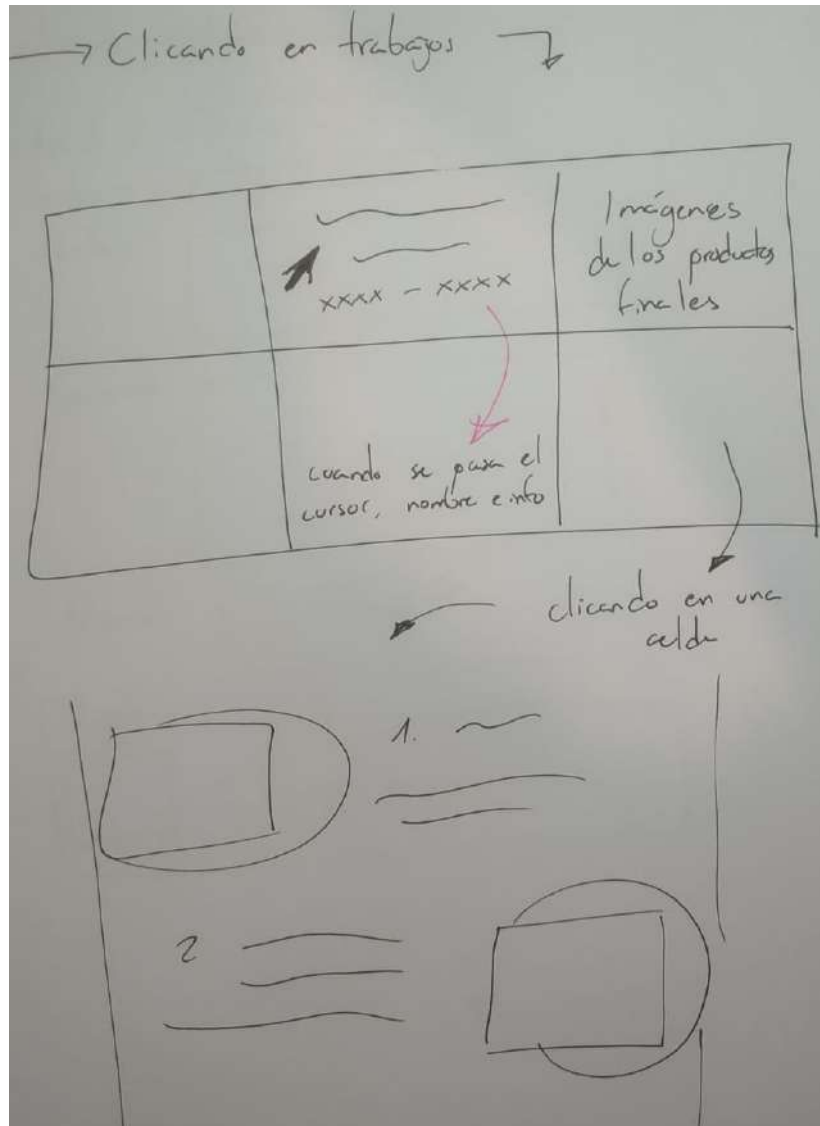


Figure 4: Catálogo de trabajos.

Clicando en *Lanza tu idea*, se pasa a una página con cajas de texto para rellenar con los datos de contacto, explicar la idea de una manera más densa y un pequeño abstract, pudiendo también adjuntar todo tipo de archivos. Estas aportaciones se pueden guardar en las cookies, para que después se pueda usar la parte inferior como un chat entre el cliente y los consultores de la empresa.

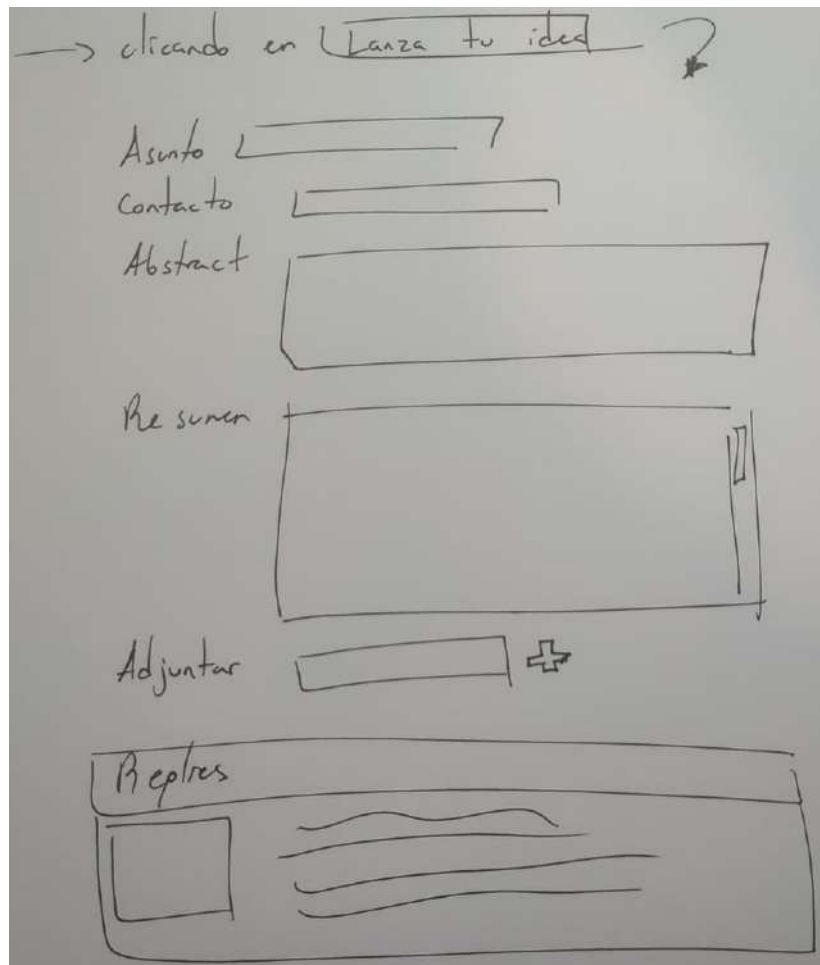


Figure 5: "Lanza tu idea".

## 5 Mockup's

Las siguientes imágenes corresponden a los mockups's ya más aproximados al diseño final de tres páginas, primero la portada (logo no definitivo), en la que en el fondo estarán pasando imágenes relacionadas, se desplegarán los menús de navegación cuando se pase el ratón, y donde debajo se presentarán los últimos aportes a la página.



Figure 6: Mockup de la Portada.

Navegando a través del apartado de trabajos se pasará a un catálogo con celdas, una para cada trabajo. Pasando el ratón se despliega el nombre del proyecto y demás información, si existen los recursos estas imágenes también estarán en movimiento.

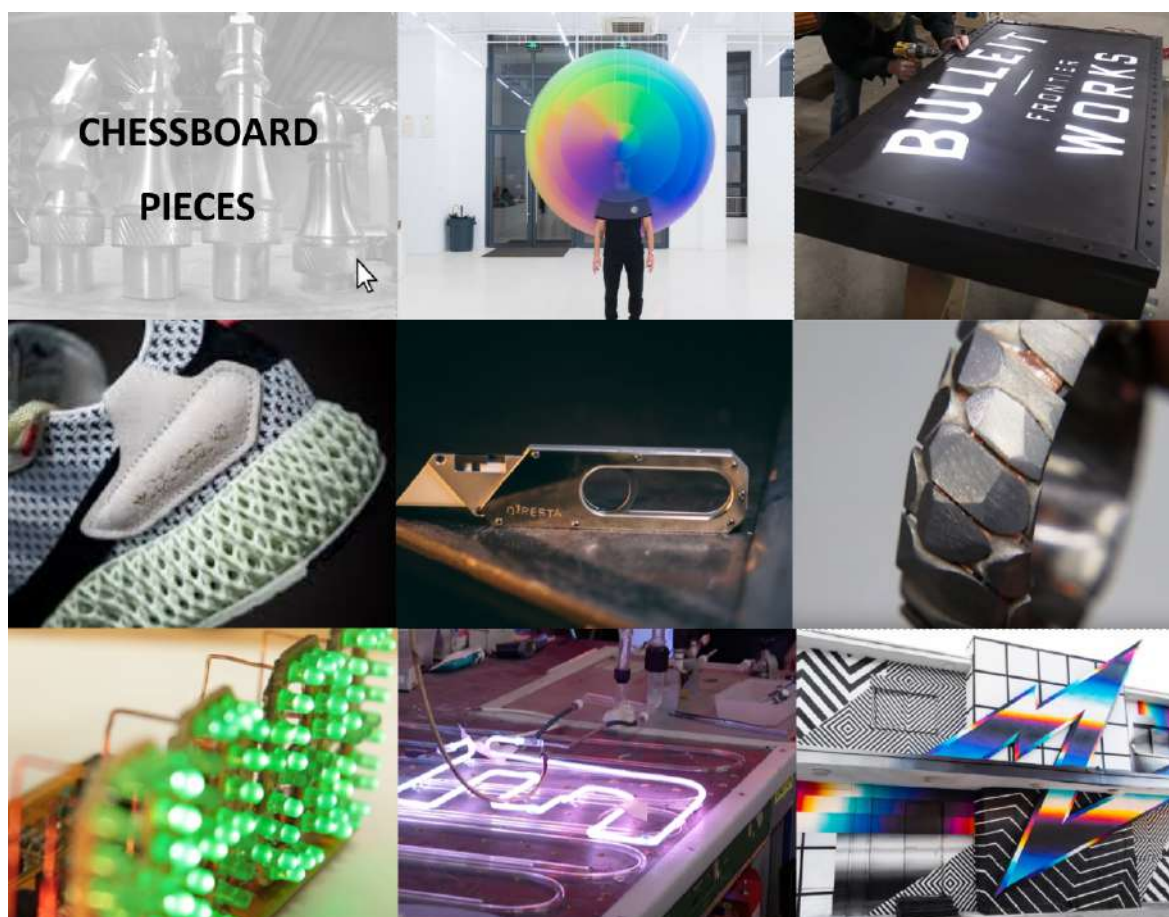
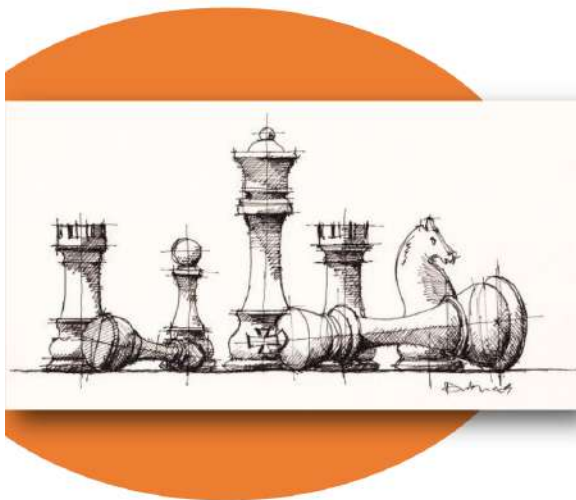


Figure 7: Mockup del listado de proyectos.

Clicando en uno de las celdas se pasa a la página del proyecto donde se explican los pasos que se siguieron y demás información de importancia que ayude a convencer a potenciales clientes (ejemplo muy rudimental, con párrafos de ejemplo).





## 1.Exposición de la idea

Pero ¿de verdad es así? La respuesta tiene que ser negativa. La política no se hace en las urnas. El sistema de partido hegemónico que impera en México desde 1929 determina que la verdadera elección de diputados haya ocurrido en febrero, probablemente cuando los diversos intereses que .

## 2. Torno

Pero ¿de verdad es así? La respuesta tiene que ser negativa. La política no se hace en las urnas. El sistema de partido hegemónico que impera en México desde 1929 determina que la verdadera elección de diputados haya ocurrido en febrero, probablemente cuando los diversos intereses que es aglutinan en el PRI acordaron las candidaturas, en un proceso que sólo es formalmente interior, pero que por la naturaleza del partido es el que verdaderamente decide la composición de la Cámara.



## 3. Detalles

Pero ¿de verdad es así? La respuesta tiene que ser negativa. La política no se hace en las urnas. El sistema de partido hegemónico que impera en México desde 1929 determina que la verdadera elección de diputados haya ocurrido en febrero, probablemente cuando los diversos intereses que es aglutinan en el PRI acordaron las candidaturas, en un proceso que sólo es formalmente interior, pero que por la naturaleza del partido es el que verdaderamente decide la composición de la Cámara.

## 4. Producto final

Pero ¿de verdad es así? La respuesta tiene que ser negativa. La política no se hace en las urnas. El sistema de partido hegemónico que impera en México desde .





## 6 Primer prototipo - HTML

Esta parte corresponde a la segunda práctica de la materia, se pretende construir con HTML el primer prototipo de la propuesta presentada anteriormente.

### 6.1 Mapa de etiquetas

En la siguiente imagen podemos ver las etiquetas de la página principal; primero un encabezado (que se repite en todas las páginas) con el logo y el nombre de la empresa, a continuación una línea horizontal provisional (será eliminada), que sirve para ver mejor las divisiones de las partes deseadas en cada página.

Más abajo en el apartado de navegación, tenemos unas listas desordenadas y anidadas, que contienen todos los enlaces necesarios para moverse por el sitio web.

Por último tenemos el apartado de Últimos aportes, donde se despliegan las imágenes de portada de los proyectos pertenecientes al portfolio de la empresa, que fueron añadidos a la web recientemente.

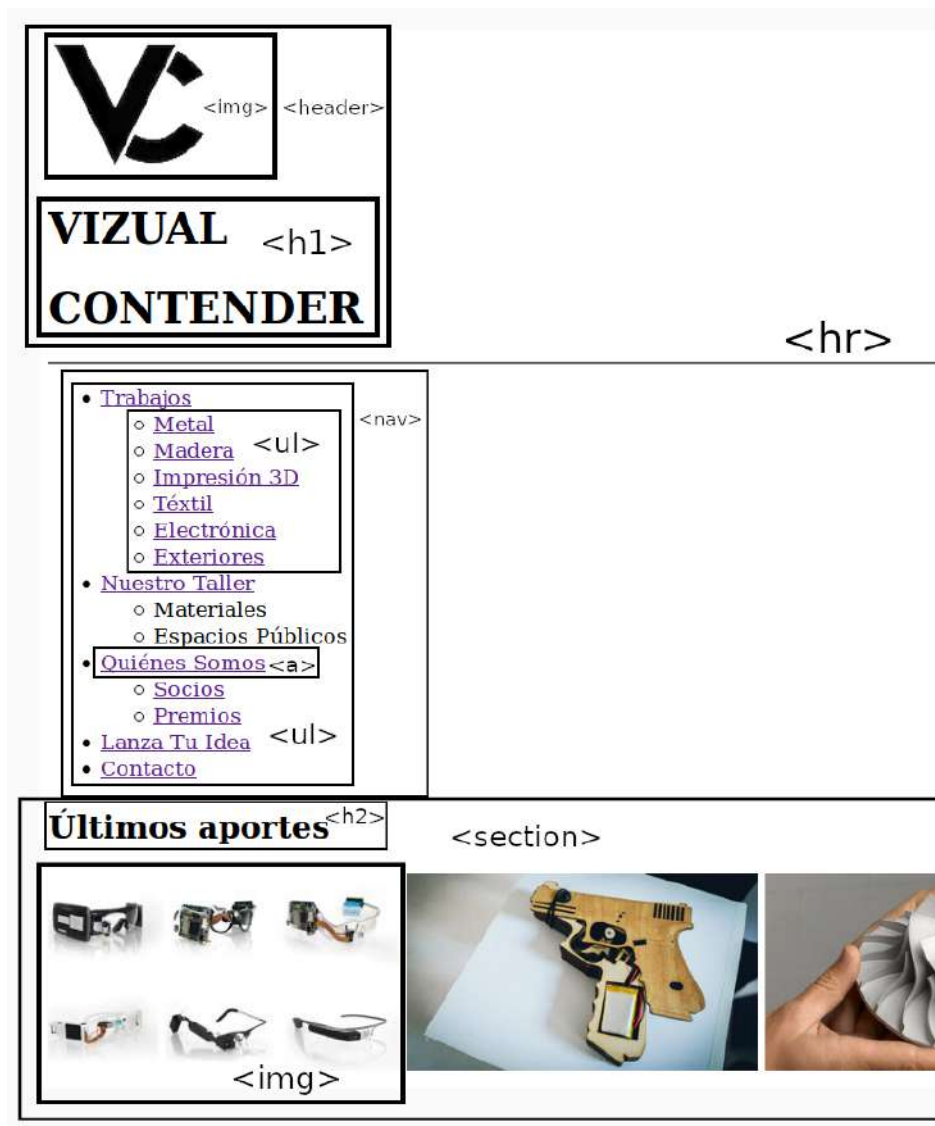


Figure 9: Mapa de etiquetas de la página principal.

En la sección **Lanza Tu Idea** los potenciales clientes son capaces de rellenar los formularios, en la parte superior están los datos de contacto para que la empresa pueda responder sin problema, más abajo donde pueden exponer su idea, presentando un título, un pequeño resumen y una explicación más extensa y detallada, además de poder adjuntar ficheros audiovisuales de todos los formatos.



## VIZUAL CONTENDER

### Lanza Tu Idea

Información de Contacto

Nombre:

E-mail:

Teléfono:

Ubicación:

<fieldset>

Desarrolla tu idea

Asunto:

<input type="text">

Abstract:

Idea:

Adjuntar:

Browse...

No file selected.

<input type="file">

Submit

<input type="submit">

### Chat/Replies

Figure 10: Mapa de etiquetas del apartado "Lanza tu Idea".

## 6.2 Estructura de despliegue

Los ficheros se estructuran jerárquicamente, con una carpeta para cada página, incluyendo una carpeta para las imágenes (a cada archivo de html principal se le llama index.html).

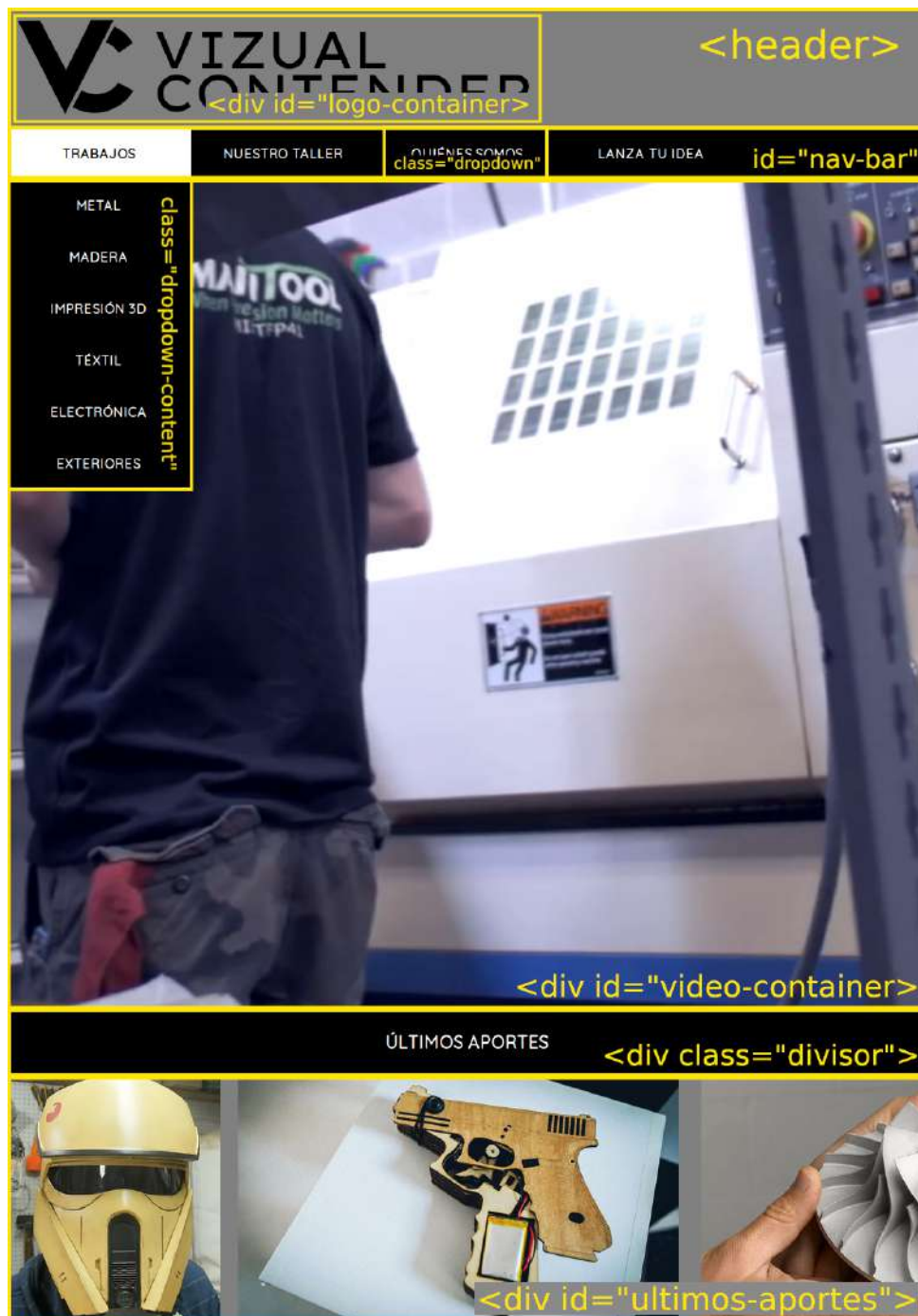
```
./mosteiroDelPilar/  
├── contacto  
│   ├── _imagenes  
│   │   ├── facebook.png  
│   │   ├── ...  
│   │   └── youtube.png  
│   ├── index.html  
│   └── proper-styles.css  
├── creacionBD.sql  
├── enObras.html  
│   ├── _imagenes  
│   │   ├── 01.png  
│   │   └── ...  
│   ├── index.html  
│   ├── lanzaTuldea  
│   │   ├── index.html  
│   │   ├── posiblesRespuestas.xml  
│   │   └── proper-styles.css  
│   └── librerias  
│       ├── angular-css.js  
│       ├── angular.js  
│       ├── angular-route.js  
│       ├── bootstrap.css  
│       ├── bootstrap.js  
│       ├── fuentes  
│       ├── jquery.js  
│       ├── popper.js  
│       └── w3.css  
├── mosteiroDelPilar.pdf  
├── nuestroTaller  
│   └── index.html  
├── portada.html  
├── proper-styles.css  
├── quienesSomos  
│   ├── _imagenes  
│   │   ├── 01.jpg  
│   │   ├── ...  
│   │   ├── premios  
│   │   └── socios  
│   ├── index.html  
│   └── proper-styles.css  
├── styles.css  
├── tienda  
│   ├── inicio.jsp  
│   ├── login.jsp  
│   ├── proper-styles.css  
│   └── validarId.jsp  
├── trabajos  
│   ├── electronica  
│   │   └── index.html  
│   ├── exteriores  
│   │   └── index.html  
│   ├── _imagenes  
│   │   ├── arduino1.jpg  
│   │   ├── ...  
│   │   └── textil.jpg  
│   ├── impresion3d  
│   │   ├── cascoRogueOne.html  
│   │   ├── _imagenes  
│   │   │   ├── ...  
│   │   └── index.html  
│   │   └── proper-styles.css  
│   ├── index.html  
│   ├── madera  
│   │   └── index.html  
│   ├── metal  
│   │   └── index.html  
│   ├── proper-styles.css  
│   ├── textil  
│   │   └── index.html  
│   └── ultimosAportes.json  
├── WEB-INF  
│   ├── classes  
│   │   ├── Cliente.class  
│   │   ├── ClienteDAO.class  
│   │   ├── ClienteDAO.java  
│   │   ├── Cliente.java  
│   │   ├── FullSignInServlet.class  
│   │   ├── FullSignInServlet.java  
│   │   ├── LogInServlet.class  
│   │   ├── LogInServlet.java  
│   │   ├── SignInServlet.class  
│   │   └── SignInServlet.java  
│   ├── creacionBD.sql  
│   ├── lib  
│   │   ├── mysql-connector-java-5.1.48-bin.jar  
│   │   └── mysql-connector-java-5.1.48.jar  
│   └── web.xml
```

## 7 Formateo y maquetación - CSS

En cuanto a la tercera práctica de la materia, esta se corresponde con la incorporación de las hojas de estilo en nuestro sitio web, empleando el lenguaje CSS.

### 7.1 Estructuras de divisiones de página

Primero la página principal o portada, en la parte superior el encabezado con el nombre de la empresa, la barra de navegación con desplegables al pasar el ratón. Después un vídeo representativo, seguido de una galería con los últimos proyectos añadidos a la página.



Otra de las páginas es la de información interna de la empresa, que divide su contenido en tres: quiénes somos, socios y reconocimientos; en el primer apartado hay un div para cada persona, en el segundo uno para cada empresa, y en el tercero se divide entre un apartado en la izquierda con el logo de la competición, y uno a la derecha con los premios ganados en ella.



TRABAJOS

NUESTRO TALLER

QUIÉNES SOMOS

LANZA TU IDEA

CONTACTO

id="our-team"

Maestra costurera.



Santiago Ferrer

Gestión de calidad.



class="persona"

Isabel Segura

Comercial de empresas de materiales



NUESTROS SOCIOS

id="socios"

class="empresa"

CORTALIA®

Técnicas Metalúrgicas

Industria Técnica Metalurgica SCCL

La fabricación, industria y comercialización de artículos de cerrajería y...



Industria Metalurgica Caesdesur SL

Forja, estampado, corte troquelado, repulsado y embutición de metales ferreos, no ferreos y aleaciones; así como la carpintería metálica

PREMIOS Y RECONOCIMIENTOS

id="reconocimientos"



PREMIOS INTERNACIONALES DE DISEÑO, BRANDING Y COMUNICACIÓN

class="competicion"

CLAP Platinum Mejor diseño de mobiliario o iluminación interior

CLAP Platinum Mejor diseño para exhibición comercial o arquitectura efímera (stands)

CLAP Platinum Mejor concepto de diseño, prototipos con alto valor innovador

class="premios"



A'Design Award and Competition - Packaging Design Category, 2018 - 2019

13



Por último la página con información de contacto, con el header y la barra de navegación, después dos primeros apartados con información y, según el lado, con distinto estilo y alineación. Después un elemento de la clase divisor que se encuentran por todo el sitio web, y la información de las tres sucursales de la empresa, con la dirección y un elemento de Google Maps. Por último un apartado con las redes sociales.





## 8 Dinamismo - JavaScript

### 8.1 Diseño responsivo

Para lograr un diseño responsivo se empleó la librería Bootstrap, en su versión 4. En el siguiente código se pueden ver las clases y la estructura empleada en la barra de navegación.

```
1 <nav class="navbar navbar-expand-lg navbar-dark">
2   <button class="navbar-toggler" type="button" data-toggle="collapse" [...] >
3     <span class="navbar-toggler-icon"></span>
4   </button>
5
6   <div class="collapse navbar-collapse" id="navbarNavDropdown" style="height: 100%">
7     <ul class="navbar-nav nav-fill">
8       <li class="nav-item">
9         <a class="nav-link" href="/index.html">Inicio</a>
10      </li>
11      <li class="nav-item dropdown">
12        <a class="nav-link dropdown-toggle" [...] >
13          Trabajos
14        </a>
15        <div class="dropdown-menu animate slideIn">
16          <a class="dropdown-item" [...] Metal </a>
17          [...]
18        </div>
19      </li>
```



Los siguiente apartados están compuestos de filas y columnas, utilizando las clases row y col para obtener los resultados responsivos, gracias a Bootstrap se colocan unas encima de otras en dispositivos con pantallas más estrechas.

```

1 <div class="container" id="our-team">
2   <div class="row">
3     <div class="col-lg-4">
4       <h3> Fernando Mosteiro </h3>
5       <p> Maquetador profesional. </p>
6       
7     </div>
8     <div class="col-lg-4"> [...] </div>
9     <div class="col-lg-4"> [...] </div>
10  </div>

```

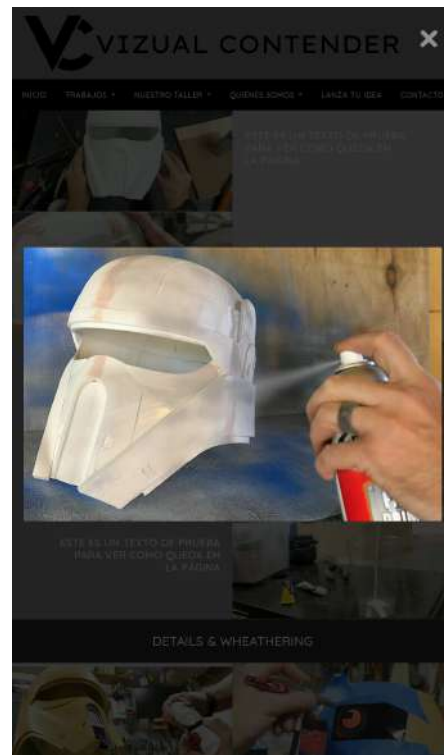




## 8.2 JQuery

En las páginas de los proyectos es posible ver más grande una imagen o un video haciendo clic en el, así este se verá cubriendo la mayoría de la pantalla para poder apreciar mejor los detalles. Debido a que producía distracciones en la atención del usuario se ponen en pausa los vídeos que se seguirían reproduciendo por detrás, y cuando se cierra la imagen se vuelve a reproducir.

```
1 function pauseVideos(bool) {
2     var x = document.getElementsByTagName("video");
3
4     for (i = 0; i < x.length; i++) {
5         if (bool) { x[i].pause(); }
6         else {     x[i].play(); }
7     }
8 }
9
10 function modalFunctionImg(element) {
11     pauseVideos(true);
12     document.getElementById("modalImg").src = element.src;
13     document.getElementById("modal").style.display = "block";
14     [...]
15 }
16
17 function closeModal() {
18     document.getElementById("modal").style.display = "none";
19     [...]
20     pauseVideos(false);
21 }
```



## 8.3 DOM

En la sección **Lanza Tu Idea** los usuarios pueden llenar los formularios con su información de contacto y la explicación de su Idea, cuando presiona el botón se envía se crea un componente en el que se incluye toda esta información, y se irán incorporando los mensajes que se intercambien en ese apartado.

```
1 $("form").submit(function (e) {
2   e.preventDefault();
3
4   var card = $(document.createElement('div')).addClass("card");
5
6   var cardBody = $(document.createElement('div')).addClass("card-body");
7   card.append(cardBody);
8
9   var titulo = $(document.createElement('h5')).addClass("card-title");
10  titulo.append($("#inputTitulo").val());
11  cardBody.append(titulo);
12  [...]
13  var dt = new Date();
14  var time = "Subido el " + dt.getDate() + "/" + (dt.getMonth()+1) + "/" +
15            dt.getFullYear() + " a las " + dt.getHours() + ":" + dt.getMinutes();
16  [...]
17  $("#chat-replies").prepend(card);
18 });
```

LANZA TU IDEA

Información de contacto

Nombre

Fernando Mosteiro

Email

fernandoMosteiro@gmail.com

Telefono

698698698

Dirección completa

Calle de Santiago de Chile, N 69, 15701, A Coruña

Desarrolla tu idea

Título

Ejemplo de idea

Abstract

Una idea (del griego *lídea* 'aspecto, apariencia, forma') relacionado con *εἶδος* eidos, 'vista, visión, aspecto') es una representación mental que surge a partir del razonamiento o de la imaginación de una persona.

idea completa

It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here, content here', making it look like readable English. Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, and a search for 'lorem ipsum' will uncover many web sites still in their infancy. Various versions have evolved over the years, sometimes by accident, sometimes on purpose (injected humour and the like).

He leído y acepto los términos y condiciones.

ENVIAR

CHAT/REPLIES

He leído y acepto los términos y condiciones.

ENVIAR

CHAT/REPLIES

EJEMPLO DE IDEA

Fernando Mosteiro

fernandoMosteiro@gmail.com

698698698

Calle de Santiago de Chile, N 69, 15701, A Coruña

Una idea (del griego *λίδα* 'aspecto, apariencia, forma') relacionado con *εἶδος* eidos, 'vista, visión, aspecto') es una representación mental que surge a partir del razonamiento o de la imaginación de una persona.

It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here, content here', making it look like readable English. Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, and a search for 'lorem ipsum' will uncover many web sites still in their infancy. Various versions have evolved over the years, sometimes by accident, sometimes on purpose (injected humour and the like).

Subido el 19/10/2019 a las 7:18

19



## 9 AJAX y AngularJS - Cargando datos de ficheros externos (XML y JSON)

### 9.1 Single Page Application

Dado como estaba estructurada la página, y al tener que copiar la barra de navegación en cada una de las páginas, lo más indicado era implementar una aplicación de página única. En el siguiente código se presenta la nueva estructura en la página index.html.

```
1 <div class="ng-view"></div> // clase de AngularJS en la que se inserta el contenido que
  cambia
2
3 <script>
4   var app = angular.module("myApp", ["ngRoute", "angularCSS"]); // Modulo AngularJS
5   app.config(function($routeProvider) {
6     $routeProvider // $routeProvider redirige el contenido
7       .when("/", { // ruta por defecto
8         templateUrl : "portada.html", // pagina de portada
9         css : "proper-styles.css" // linkeando el css correspondiente
10      })
11      .when("/impresion3d", {
12        templateUrl : "trabajos/impresion3d/cascoRogueOne.html",
13        css : "trabajos/impresion3d/proper-styles.css"
14      })
15
16      [El resto de las paginas...]
17   });
18 </script>
```

Por lo que sólo declaramos el `< head >`, linkeamos los CSS's y las librerías una vez en index.html. Así el header se comparte en todas las páginas, y todo lo inferior es cambiante.





## 9.2 Cargando fichero JSON

En la página de portada, en la parte inferior, existe un apartado con los últimos aportes subidos a la web. El plan es que una consulta a la base de datos devuelva la información de los aportes en un archivo JSON, ordenados por la fecha más reciente.



El archivo JSON contendrá la siguiente información, para cada aporte:

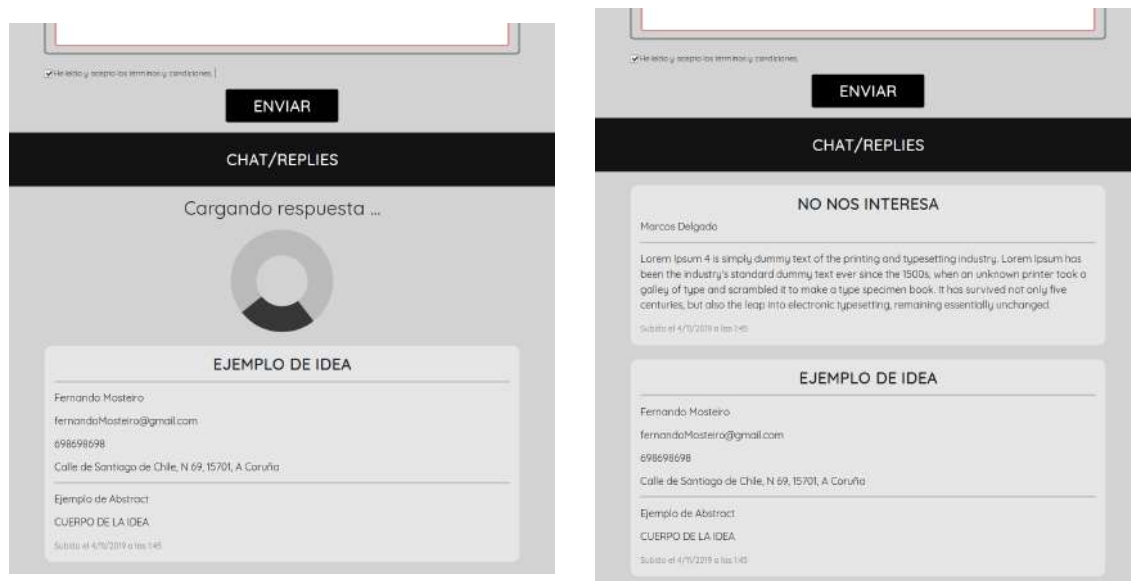
```
1 {"Aportes": [  
2   {"enlace": "#!impresion3d",  
3     "imgSrc": "_imagenes/01.png",  
4     "titulo": "Coastal <br> Defender <br> Stormtrooper <br> Helmet",  
5     "fecha": ""},  
6  
7   [Otros aportes...]
```

Y el contenido HTML es generado a través de la siguiente función:

```
1 $.getJSON("ultimosAportes.json", function (data) { // Funcion de JQuery que lee el JSON  
2   $.each(data.Aportes, function (index, aporte) { // Iterando por cada aporte que se lee  
3     var A = document.createElement("a");  
4     A.setAttribute("href", aporte.enlace);      // se crea su enlace  
5  
6     var I = document.createElement("img");  
7     I.setAttribute("src", aporte.imgSrc);      // se crea la imagen con su  
8       correspondiente src  
9     A.append(I);                               // se anade al enlace  
10  
11     var texto = document.createElement("p");  // se crea y se anade el mensaje  
12     texto.innerText = aporte.titulo;  
13     A.append(texto);  
14     $(".aportes").append(                     // se anade a la seccion aportes  
15       $(document.createElement("figure")).append(A) // dentro de una figura  
16     )  
17   });  
18 });
```

### 9.3 Cargando fichero XML

En el apartado **Lanza tu idea**, el usuario puede recibir una de las respuestas generadas automáticamente (de momento), poco después de enviar su mensaje. De la siguiente manera:



Las respuestas están contenidas en el fichero XML de la siguiente manera:

```
1 <POSIBLESRESPUESTAS>
2
3   <RESPUESTA>
4     <TITULO> Propuesta aceptada </TITULO>
5     <AUTOR> Fernando Mosteiro </AUTOR>
6     <CUERPO>
7       Lorem Ipsum 1 is simply dummy text of the printing and typesetting industry.
8       Lorem Ipsum has been the industry's standard dummy text ever since the 1500s
9       [...]
10    </CUERPO>
11  </RESPUESTA>
12
13  <RESPUESTA>
14    <TITULO> Propuesta rechazada </TITULO>
15    <AUTOR> Blas Abalde </AUTOR>
16    <CUERPO>
17      Lorem Ipsum 2 is simply dummy text of the printing and typesetting industry.
18      Lorem Ipsum has been the industry's standard dummy text ever since the 1500s
19      [...]
20    </CUERPO>
21  </RESPUESTA>
22
23  [Otras respuestas ...]
24
25 </POSIBLESRESPUESTAS>
```

El archivo es interpretado por la siguiente función, empleando la clase XMLHttpRequest :

```
1 function cargaRespuesta() {
2     var xhttp=new XMLHttpRequest();
3
4     xhttp.onreadystatechange = function() {
5         if(this.readyState===4 && this.status===200) {
6             var xmlDoc = this.responseXML;
7
8             var t = xmlDoc.getElementsByTagName("TITULO");
9             var a = xmlDoc.getElementsByTagName("AUTOR");
10            var c = xmlDoc.getElementsByTagName("CUERPO");
11
12            // se selecciona una de las respuestas de manera aleatoria
13            var indexRespuesta = Math.floor(Math.random() * t.length);
14
15            var card = $(document.createElement('div')).addClass("card");    // Se crean los
16                componentes html
17            var cardBody = $(document.createElement('div')).addClass("card-body");
18            card.append(cardBody);
19
20            var titulo = $(document.createElement('h5')).addClass("card-title");
21            titulo.append(t[indexRespuesta].childNodes[0].nodeValue); // se selecciona el titulo
22                del vector con todos
23            cardBody.append(titulo);
24
25            [De manera similar con el autor y el cuerpo ...]
26
27            $(document.getElementById("cargando")).remove();    // Se elimina el icono de carga
28                de respuesta
29
30            $("#chat-replies").prepend(card);    // Se anade en las respuestas
31        }
32    };
33 }
```

## 10 Logueo y registro de usuarios en una BD mediante formularios

### 10.1 Log In

A continuación se presenta lo primero que el usuario ve al acceder a la apartado de la tienda, donde puede loguearse si ya está registrado, o crear un nuevo usuario completando los formularios correspondiente.

Al presionar el botón **LOG IN**, se pasa al primero de los tres servlets, que comprueba si el usuario está registrado en la base de datos del alumno, y si su contraseña es la correcta. Si alguna de estas dos condiciones no se cumple, se muestra el pequeño mensaje de error en la parte inferior.

Formulario de identificación con el título "IDENTIFÍCATE". Contiene dos campos de entrada: "E-mail" y "Contraseña". Debajo de los campos hay dos botones: "LOG IN" y "SIGN IN".

Formulario de identificación con el título "IDENTIFÍCATE". Muestra un mensaje de error en rojo: "E-MAIL Y/O CONTRASEÑA INCORRECTOS". Los campos de entrada y los botones "LOG IN" y "SIGN IN" están presentes.

```
1 public void doPost(HttpServletRequest request, HttpServletResponse response) throws
  ServletException, IOException {
2   String email = request.getParameter("LogInEmail"), pass =
    request.getParameter("LogInPassword"), respuesta = ""; // obteniendo los datos de los
    formularios
3
4   Cliente c = new Cliente(email, pass); // Objeto cliente, usado para las funciones de SQL
5
6   if (clienteDAO.validarLogin(c)) { // comprueba en la BD del alumno
7     // mensaje de bienvenida y pasa a la tienda
8     request.getSession().setAttribute("bienvenidaTienda", "Bienvenido/a de nuevo!");
9     response.sendRedirect("/VizualContender/#!tienda");
10  } else {
11    // mensaje de error y vuelve al login
12    request.getSession().setAttribute("mensajeLogin", "E-mail y/o contrasea incorrectos");
13    response.sendRedirect("/VizualContender/#!login");
14  }
15 }
```

```

1 public boolean validarLogin(Cliente cliente) {
2     PreparedStatement preparedStatement;
3     ResultSet result;
4
5     try {
6         // funcion de coneccin
7         Connection connection = conectandoBD("mysql", "127.0.0.1", "3306",
8             "mosteiroDelPilar", "daw", "daw");
9
10        //consulta SQL
11        String query= "SELECT * FROM usuario WHERE email=? AND contrasena=?";
12
13        preparedStatement = connection.prepareStatement(query);
14
15        preparedStatement.setString(1, cliente.getEmail());    // poniendo los valores en la
16        consulta
17        preparedStatement.setString(2, cliente.getContrasena());
18
19        result = preparedStatement.executeQuery();
20
21        return result.next(); // booleano si se recupero alguna fila
22    } catch(Exception e) {
23        System.out.println(e.getMessage());
24    }
25 }

```

## 10.2 Sign In

Al clicar en el botón **SIGN IN** pasamos al siguiente apartado (ver captura), donde el nuevo usuario debe de insertar todos sus datos, siendo los correspondientes a la dirección de envío, y el método de pago opcionales.

En la figura del centro, vemos una demostración con todos los inputs con la propiedad pattern, que son chequeados a través de expresiones regulares, como pueden ser el DNI o el código postal.

- DNI = "[0-9]{8}[A-Za-z]{1}"
- Número = "\d+"
- Código postal = "\d{5}"
- Número de tarjeta = "[0-9]{16}"
- Caducidad = "\d{2}/\d{2}"
- CVV = "\d{3}"

Además el formulario es capaz de, a través de JavaScript, deshabilitar los campos de los datos de la tarjeta cuando el método de pago es uno distinto.

The image shows three sequential screenshots of a web form titled 'REGISTRATE'. The form is divided into three main sections: 'DIRECCIÓN DE ENVÍO' (Shipping Address) and 'MÉTODO DE PAGO' (Payment Method).  
 - The first screenshot shows the form with empty input fields.  
 - The second screenshot shows the form filled with sample data: Name 'Manuel', Surname 'Lestón Lestón', Email 'manuleston@gmail.com', DNI '49468V', Address 'Calle Rosalia de Castro', City 'Santiago', Province 'A coruña', and Payment Method 'Tarjeta'.  
 - The third screenshot shows the form filled with real data: Name 'Manuel', Surname 'Lestón Lestón', Email 'manuleston@gmail.com', DNI '49465858V', Address 'Calle Rosalia de Castro', City 'Santiago', Province 'A coruña', and Payment Method 'Tarjeta'.  
 Each screenshot has a 'CONTINUAR' button at the bottom.

Presionando en **CONTINUAR** se comprueba en la base de datos del alumno si el email ya está registrado, en caso afirmativo el usuario vuelve a la ventana de login con un mensaje indicando el error, de manera similar a la mostrada anteriormente.

En caso de que el correo no esté ya registrado, el servlet recupera la información de los formularios como parámetros, donde almacena la información pertinente en la base de datos de profesor y recupera el id y lo envía como cookie, y el resto de datos los almacena como atributos en la siguiente página, que es la de validación del Id.

```

1 public void doPost(HttpServletRequest request, HttpServletResponse response) throws
  ServletException, IOException {
2     Cliente c = new Cliente(
3         0, // id, que aun no tenemos asegurado
4         request.getParameter("signinNombre"), // nombre
5         [...], // etc.
6         request.getParameter("cvvTarjeta")); // cvv de la tarjeta de credito
7
8     if (clienteDAO.existAlready(c)) {
9         // mensaje de error y vuelta al login
10        Session().setAttribute("mensajeSignin", "El correo ya esta registrado");
11        response.sendRedirect("/VizualContender/#!login");
12    } else {
13        int id = clienteDAO.insertBDprofe(c); // inserta en la Bd y devuelve en nuevo ID
14
15        Cookie c1 = new Cookie("mosteiroDelPilar", String.valueOf(id));
16        c1.setMaxAge(60*5);
17        response.addCookie(c1); // Objeto cookie y se envia
18
19        // Atributos con la info que pasa a la siguiente pagina
20        request.getSession().setAttribute("nombreRegistro", c.getNombre());
21        [...];
22        request.getSession().setAttribute("cvvPago", c.getCvv());
23
24        response.sendRedirect("/VizualContender/#!validarId"); // siguiente pagina
25    }
26 }

```



Ahora el usuario pasa a la parte donde simulamos que se le notifica un código de confirmación. Se muestra un listado con la información que va a ser registrada en la base de datos del alumno, además de un formulario donde debe introducir el id que se le muestra la cookie correspondiente al id en la alerta SMS, y su contraseña confirmada.

**Tienes un SMS!**

Valida tu registro, inserta el id que se te ha enviado.

id

Contraseña

Repita contraseña

**ACEPTAR**

---

**DATOS DEL REGISTRO**

Nombre: Manuel	Dirección de facturación la misma que la de envío: true
Apellidos: LestÃ³n LestÃ³n	Método de pago: tarjeta
Correo: manuleston@gmail.com	Numero de tarjeta: 1234567890123456
DNI: 49465858V	Fecha de caducidad: 15/22
	CVV: 911
Calle: Calle Rosalia de Castro	Ciudad: Santiago
Núm.: 15	Provincia: A coruña
Piso: 1H	CP: 15701

**Tienes un SMS!**

Tu ID correspondiente es: 8

**OK**

Contraseña

Repita contraseña

La cookie es recuperada a través del siguiente código, usando Java en un archivo .jsp; y mostrada usando una `alert()`.

```

1 | <% String miCookie = "YOU LATE";
2 |     Cookie[] cookies = request.getCookies();
3 |     for(int i = 0; i < cookies.length; i++) { // iterando por las cookies
4 |         Cookie c = cookies[i];
5 |         if(c.getName().equals("mosteiroDelPilar")) { // la mia
6 |             miCookie = c.getValue();
7 |             break;
8 |         }
9 |     } %>
10 |
11 | <script>
12 |     // el "sms"
13 |     $("button#sms").click( function() { alert("Tu ID correspondiente es: <%=miCookie%>"); });
14 | </script>

```

Además se hace un chequeo cada vez que se escribe algo en los campos de las contraseñas, y si los valores no coinciden se deshabilita el botón de **Aceptar**. El id que el usuario introduzca es verificado con el valor de la base de datos del profesor, y si no coincide se muestra un pequeño mensaje similar a los anteriores (ver Figura). En caso de coincidir el id, la contraseña y el resto de datos que se trajeron desde el anterior formulario, son insertados en la base de datos del alumno.

Valida tu registro, inserta el id que se te ha enviado.

---

Las contraseñas no coinciden!

**ACEPTAR**

Valida tu registro, inserta el id que se te ha enviado.

---

**ACEPTAR**

DATOS DEL REGISTRO

```

1 <script>
2   //cada vez que cambien las contraseas
3   $("#contrasena, #repitaContrasena").keyup( function() {
4     // comprueba los valores
5     if ($("#contrasena").val() == $('#repitaContrasena').val()) {
6       // se quita el mensaje y se activa el boton
7       $("#mensajeContrasena").css("display", "none");
8       $("#botonSubmit").attr("disabled", false);
9     } else {
10      // se pone el mensaje y se desactiva el boton
11      $("#mensajeContrasena").css("display", "block");
12      $("#botonSubmit").attr("disabled", true);
13    }
14  });
15 </script>

```

En el siguiente código vemos la inserción de los datos del usuario en la base de datos del alumno. Al insertar al nuevo usuario en la base de datos del alumno, pasamos a la página de inicio del apartado **Tienda** de nuestra página, donde mostramos un mensaje de bienvenida con el nombre del nuevo usuario.

```
1 public void insertBDalumno(Cliente cliente) {
2     PreparedStatement preparedStatement;
3
4     try {
5         // conexion
6         Connection connection = conectandoBD("mysql", "127.0.0.1", "3306",
7             "mosteiroDelPilar", "daw", "daw");
8
9         String query = "INSERT INTO usuario VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";
10
11         preparedStatement = connection.prepareStatement(query);
12
13         // insertando la informacion, desde el objeto cliente
14         preparedStatement.setInt(1, cliente.getId());
15         [...]
16         preparedStatement.setString(17, cliente.getCvv());
17
18         preparedStatement.executeUpdate();
19     } catch (Exception e) {
20         System.out.println(e.getMessage());
21     }
22 }
```





## 11.2 Catálogo de artículos

Después de validar el login del usuario, se obtienen las instancias de las clases java con la información almacenada en la base de datos.

```
public Usuario getUsuarioBD(String email) {
    PreparedStatement preparedStatement;
    ResultSet rs;
    Usuario usuario = null;

    try {
        String query = "SELECT * FROM usuario WHERE email=?"; // consulta sql

        preparedStatement = this.getConexion().prepareStatement(query);
        preparedStatement.setString(1, email);
        rs = preparedStatement.executeQuery();

        if (rs.next()) {
            // crear usuario
            usuario = new Usuario(rs.getInt("id"),
                rs.getString("nombre"), rs.getString("apellidos"),
                rs.getString("email"), rs.getString("dni"),
                rs.getString("contrasena"));

            // obtener el resto de información
            usuario.setDirecciones(getFachadaBD().getDirecciones(usuario));
            usuario.setMetodosPago(getFachadaBD().getMetodosPago(usuario));
            usuario.setCarrito(getFachadaBD().getCarrito(usuario));
            usuario.setPedidos(getFachadaBD().getPedidos(usuario));
        }

        return usuario;
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }

    return usuario;
}
```

Este usuario es almacenado como un atributo en la sesión, además de almacenar la cookie "mosteiroDelPilar" con el valor "mosteiroDelPilar", que servirá como indicador de que hay una sesión iniciada.

Así se muestra la página de inicio de la tienda, que contiene la lista de artículos, leídos de la base de datos e iterando usando las etiquetas JSTL.



## NIKE LD WAFFLE SACAI BLACK

After making their debut on the Fall / Winter 2019 runway in Paris Fashion Week, Nike and Sacai deliver a familiar silhouette in new fashion with the Sacai Nike LD Waffle Black, now available on StockX. The LD Waffle took the sneaker world by storm when they originally debuted in Sacai's Spring / Summer 2019 show alongside a Nike Blazer collaboration. Out of the two sneakers they worked on together, the Waffle stood out most and acquired an aftermarket hype that hasn't showed any signs of slowing down. It only made sense for the two companies to keep this model alive, but present it in new colorways.

440.00 €

Unidades disponibles: 70

— 1 +

AÑADIR A LA CESTA



## NIKE LD WAFFLE SACAI PINE GREEN

After making their debut on the Fall / Winter 2019 runway in Paris Fashion Week, Nike and Sacai deliver a familiar silhouette in new fashion with the Sacai Nike LD Waffle Pine Green, now available on StockX. The LD Waffle took the sneaker world by storm when they originally debuted in Sacai's Spring / Summer 2019 show alongside a Nike Blazer collaboration. Out of the two sneakers they worked on together, the Waffle stood out most and acquired an aftermarket hype that hasn't showed any signs of slowing down. It only made sense for the two companies to keep this model alive, but present it in new colorways.

271.00 €

Unidades disponibles: 93

— 1 +

AÑADIR A LA CESTA



## NIKE WAFFLE RACER OFF-WHITE VIVID SKY

Lorem ipsum is simply dummy text of the printing and typesetting industry. Lorem ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

185.00 €

Unidades disponibles: 194

— 1 +

AÑADIR A LA CESTA



## NIKE WAFFLE RACER OFF-WHITE BLACK

Lorem ipsum is simply dummy text of the printing and typesetting industry. Lorem ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

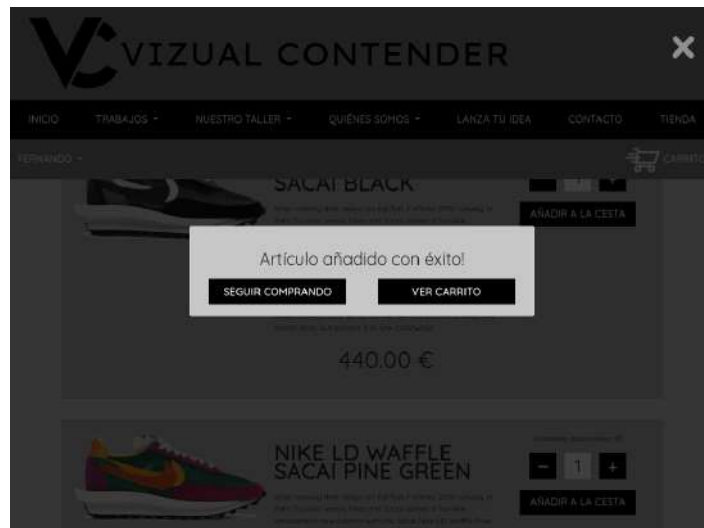
212.00 €

Unidades disponibles: 7

— 1 +

AÑADIR A LA CESTA





En el siguiente código se muestra como se itera a través de las filas en la tabla de artículos:

```
<c:choose>
  <c:when test="{articulosBD.rowCount !=0}">
    <c:forEach var="articulo" items="{articulosBD.rows}">
      <div class="row articulo">
        <div class="col-lg-4" style="...">
          
        </div>

        <div class="col-lg-5">
          <p class="nombreArticulo"> {articulo.nombre} <p>
          <p class="descripcionArticulo"> {articulo.descripcion} <p>
          <p class="precioArticulo"> {articulo.precio} € <p>
        </div>

        <div class="col-lg-3" style="...">
          <c:choose>
            <c:when test="{articulo.stock <= 10}">
              <p class="stockArticulo" style="..."> Unidades disponibles: {articulo.stock} <p>
            </c:when>
            <c:otherwise>
              <p class="stockArticulo"> Unidades disponibles: {articulo.stock} <p>
            </c:otherwise>
          </c:choose>

          <form action="{pageContext.request.contextPath}/CarritoServlet" method="post">
            <input class="idProducto" name="idArticulo" type="hidden" value="{articulo.id}">
            <input name="tipo" type="hidden" value="anadirUnidades">

            <button class="cambiarCantidad menos" type="button"> - </button>
            <input class="cantidad" name="cantidad" type="text" value="1">
            <button class="cambiarCantidad mas" type="button"> + </button>

            <button class="añadirCesta" type="submit"> Añadir a la cesta </button>
          </form>
        </div>
      </div>
    </c:forEach>
  </c:when>
  <c:otherwise>
    La base de datos está vacía.
  </c:otherwise>
</c:choose>
```



Al presionar el botón "Añadir a la cesta", se da inicio al siguiente servlet, que es usado en todos los cambios en el carrito. En él se hacen todos los cambios pertinentes en la instancia Carrito del usuario correspondiente, y luego se reflejan los cambios en la base de datos.

```
public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException {
    // usuario logueado como atributo en la sesión
    Usuario usuario = (Usuario) request.getSession().getAttribute( s: "loggedUsuario");

    // valores del artículo según el formulario seleccionado
    int articulo = Integer.parseInt(request.getParameter( s: "idArticulo"));
    int cantidad = Integer.parseInt(request.getParameter( s: "cantidad"));

    // el servlet es válido para tres tipos de cambios en el carrito
    switch (request.getParameter( s: "tipo")) {
        case "anadirUnidades":
            // añadiendo X cantidad al carrito desde el catálogo
            usuario.getCarrito().addToCarrito(fachadaBD.getArticulo(articulo), cantidad);

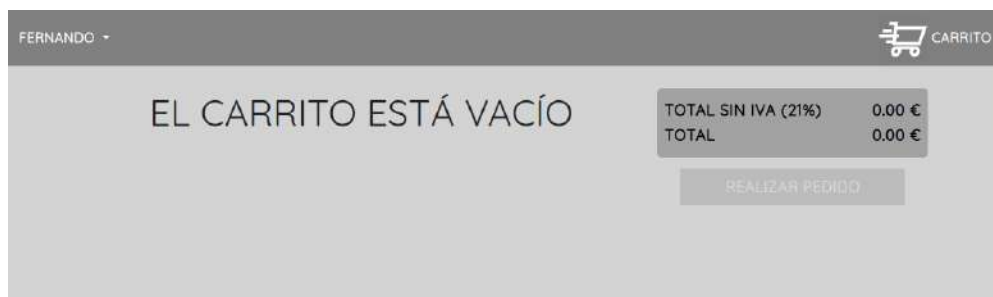
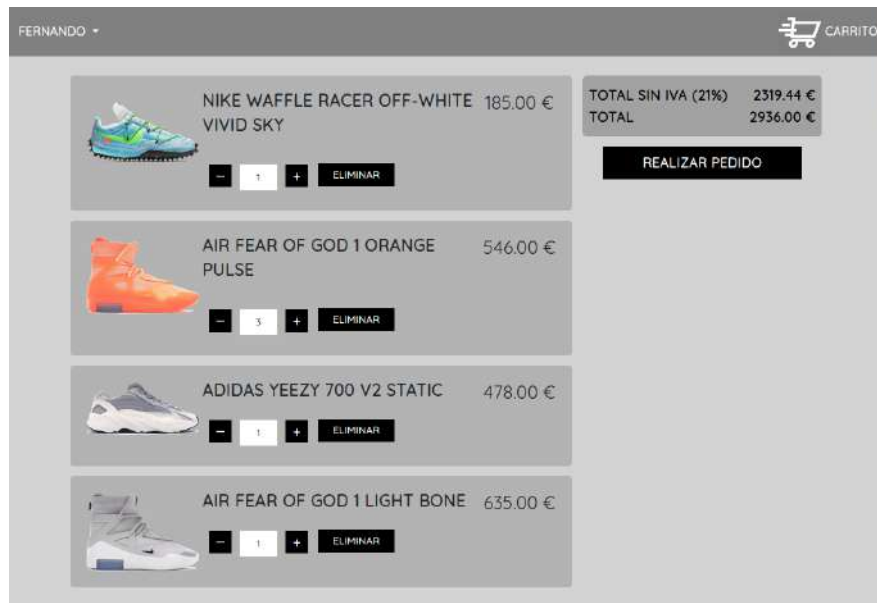
            // Cookie para mostrar mensaje de éxito
            Cookie cook = new Cookie( name: "mostrarMensaje", value: "true");
            cook.setMaxAge(3);
            response.addCookie(cook);
            break;
        case "cambiarUnaUnidad":
            // añadiendo o eliminando una unidad de un artículo en el carrito
            usuario.getCarrito().setCantidadArticulo(fachadaBD.getArticulo(articulo), cantidad);
            break;
        case "eliminarArticulo":
            // eliminando todas las unidades de un artículo en el carrito
            usuario.getCarrito().removeAllFromCarrito(fachadaBD.getArticulo(articulo));
            break;
        default:
            return;
    }

    synchronized (request) { // cambios en la base de datos
        fachadaBD.insertCarrito(usuario);
    }

    // redirigiendo a los correspondientes jsp según el tipo de cambio
    if (((String) request.getParameter( s: "tipo")).equals("anadirUnidades"))
        response.sendRedirect( s: "/mosteiroDelPilar/#!tienda");
    else
        response.sendRedirect( s: "/mosteiroDelPilar/#!carrito");
}
```

### 11.3 Carrito y completar pedido

El usuario puede ver el estado de su carrito en todo momento, al estar almacenado en la base de datos, su información se obtiene de manera similar al catálogo de artículos mediante JSTL.



En esta página el usuario también es capaz de modificar el carrito, incrementando o disminuyendo la cantidad, o eliminando el artículo completamente (todo esto usando el servlet anterior). Además se le muestra el total con y sin IVA de los artículos en el carrito, mediante los métodos del objeto.

Pulsando el botón "Realizar pedido", se muestra en una nueva página un resumen de los contenidos del carrito, además de las direcciones de envío y métodos de pago almacenados y relacionados con el usuario.

FERNANDO

CARRITO

	NIKE WAFFLE RACER OFF-WHITE VIVID SKY	1x	185.00 €
	AIR FEAR OF GOD 1 ORANGE PULSE	3x	546.00 €
	ADIDAS YEEZY 700 V2 STATIC	1x	478.00 €
	AIR FEAR OF GOD 1 LIGHT BONE	1x	635.00 €

DIRECCIONES DE ENVIO

CUERPO DE ARMAS 11

776

Urbano del Nordeste, Tarragona

43700

PROSEGUR 10

Calle de la Victoria, Valencia

46100

MÉTODOS DE PAGO

Indicaciones que se envían desde

Tipo de pago

TOTAL SIN IVA (21%)

2319.44 €

TOTAL

2936.00 €

VOLVER

CONFIRMAR PEDIDO



Al presionar el botón "Confirmar pedido" se inicia el servlet que almacena el pedido en la base de datos, con la fecha de realización, dirección y método de pago seleccionados; los artículos, sus cantidades y el total; además de vaciar el carrito del usuario.

```

public void doPost(HttpServletRequest request, HttpServletResponse response) throws Serv
    // datos de atributos de sesión y formulario
    Usuario usuario = (Usuario) request.getSession().getAttribute( s: "loggedUsuario");
    int direccion = Integer.parseInt(request.getParameter( s: "direccionEnvio"));
    int metodoPago = Integer.parseInt(request.getParameter( s: "metodoPago"));

    synchronized (request) {
        fachadaBD.insertPedido(usuario, direccion, metodoPago);
    }

    // cookie de mensaje de éxito
    Cookie cook = new Cookie( name: "mostrarMensaje", value: "true");
    cook.setMaxAge(4);
    response.addCookie(cook);

    response.sendRedirect( s: "/mosteiroDelPilar/#!pedidos");
}

```

}

contacto obtenida del fichero web.xml.

## 11.4 Usuario e información almacenada

El usuario puede consultar su información y sus pedidos realizados. Esta información es obtenida de la base de datos, por lo que también se utilizan las etiquetas JSTL.

FERNANDO +CARRITO

DATOS PERSONALES


Nombre: Fernando Mosteiro del Pilar

Correo electrónico: fernandomosteiro@gmail.com

DNI: 12345678A

DIRECCIONES DE ENVÍO

MÉTODOS DE PAGO


CARRITO

Vº de pedido: **402** Realizado el: **2019-02-15**

Vº de pedido: **301** Realizado el: **2019-02-15**

Vº de pedido: **3008** Realizado el: **2019-02-16**

**DIRECCION DE ENVÍO**






Rta Seguros 49  
Cabriles, Maricao  
9933

**METODO DE PAGO**

Funcionará para su envío

Tipo

**ARTÍCULOS**

	Nike LD Waffle Vapour Black	2x	640.00 €
	Jordan 6 Vapour Boost 380 Alien	1x	458.00 €
	Jordan 6 Vapour Boost 700 V2 Violet	1x	547.00 €
	Air Fear Of God 1 Orange Pulse	1x	546.00 €
	Jordan 6 Vapour Boost 700 V2 Static	3x	478.00 €

**TOTAL:**

**3665.00 €**

Vº de pedido: **3007** Realizado el: **2019-02-14**