

TRABAJO FINAL DE GRADO

Qualitycourses

Nueva web de la empresa

QualityCourses

www.quality-courses.com

Daniel Correa Placeres
Grado Multimedia

Directora: Ana Gabriela Zuñiga Zarate

Índice

0. Introducción y nacimiento del proyecto.....	pág. 4
0.1. Objetivos.....	pág. 6
1. Estudio de mercado	pág. 7
1.1. La empresa	pág. 8
1.2. La competencia y referencias	pág. 8
2. Diseño	pág. 12
2.1. Tipografía	pág. 15
2.2. Color.....	pág. 15
2.3. Iconos	pág. 16
2.4. Maquetación HTML.....	pág. 17
3. Base de datos	pág. 25
4. Programación backend	pág. 27
4.1. El núcleo	pág. 27
4.2. Módulos.....	pág. 34
4.2.1. Url Engine.....	pág. 34
4.2.2. Lang Engine	pág. 36
4.2.3 Lang Menu	pág. 38
4.2.4 Meta tags Module	pág. 39
4.2.5 Footer	pág. 40
4.2.6 Booking tool	pág. 41
4.3. Home	pág. 47
4.4. Ciudad/País	pág. 49
4.5. Escuela	pág. 55
4.6. Formulario inscripción	pág. 57
4.7. Checkout.php y ok.php	pág. 60
4.8. Contacto	pág. 63
4.9. Ofertas	pág. 66
4.10. Default	pág. 67

4.11. Ajax/Jquery	pág. 68
5. Usabilidad	pág. 74
5.1. Pruebas de usabilidad versión final	pág. 77
6. SEO	pág. 83
6.1. Velocidad de carga	pág. 84
6.2. Títulos	pág. 85
6.3. Contenidos	pág. 85
6.4. Metas	pág. 86
6.5. Snippets	pág. 86
6.6. Blog	pág. 87
7. Analítica web	pág. 88
7.1. Visitas y porcentaje de rebote	pág. 89
7.2. Comercio electrónico	pág. 91
7.3. Objetivos.....	pág. 92
7.4. Adquisición	pág. 92
7.5 Eventos.....	pág. 93
8. Conclusiones.....	pág. 94
8.1. Técnico.....	pág. 95
8.2. Madurez del código.....	pág. 96
8.3. Diseño.....	pág. 96
8.4. Otras áreas.....	pág. 96
8.5. Puntos débiles.....	pág. 96
8.6 Futuro de la aplicación	pág. 97
9. Bibliografía.....	pág. 98
10. Agradecimientos.....	pág. 101

0. Introducción y nacimiento del proyecto.

0.1. Objetivos

0. Introducción y nacimiento del proyecto

El proyecto nace durante la realización de las prácticas laborales en la empresa First Class Education S.L, donde llega a la empresa, proveniente de un nuevo inversor una serie de dominios para los cuales hay que crear webs que vendan nuestros productos. Es en este momento cuando nace la idea de realizar para ellos este TFG.

La idea inicial era básicamente la creación de las webs en distintos idiomas para los nuevos dominios adquiridos. Pero después de la realización de algunos test de usabilidad explicados más adelante, sobre una versión beta de estas webs, donde la intención era comparar la web principal de la empresa www.quality-courses.com en ese momento, con la nueva web creada para los nuevos dominios, se decide dar un giro al proyecto y centrar todas los recursos y tiempo en esta web principal (también en varios idiomas) y dar un cambio radical a la imagen corporativa de la empresa antes mencionada.

Con la realización de este proyecto principalmente se quiere demostrar y poner en práctica los conocimientos en varias áreas, como programación, diseño, usabilidad, seo y marketing online.

El momento en que se toma la decisión de involucrarse en este reto es después de 3 meses trabajando en la empresa pero en ese tiempo se han asumido bastantes responsabilidades y se considera que se conoce la empresa, sus puntos fuertes, sus debilidades y el producto que se vende, y es por ello que se decide apostar por el cómo trabajo final de grado.

0.1 Objetivos

El objetivo de este proyecto es demostrar y poner en práctica los conocimientos de programación, diseño, usabilidad, seo y marketing online.

El proyecto consta de una web, en varios idiomas, Castellano, Inglés, Italiano, Alemán y Francés, destinada a la venta del producto de la empresa. En este caso cursos de idiomas en el extranjero y paquetes de curso mas alojamiento, en las de 70 países y 200 escuelas de idiomas alrededor del mundo.

Al tratarse de una empresa que opera principalmente online ,el objetivo es realizar una aplicación web que de una mejor visión de la imagen de marca, genere confianza en el comprador, contenga toda la información que el cliente busca, sea usable y atractiva para el usuario.

En definitiva, el objetivo es la realización de una web que mejore la conversión y el número de ventas además de mejorar el posicionamiento web y el tráfico.

1. Estudio de mercado.

1.1. La empresa.

1.2. La competencia y referencias.

1. Estudio de mercado

1.1 La empresa

First Class Education se trata de una agencia que actúa como intermediaria entre escuelas de idiomas en más de 50 países y estudiantes de todo el mundo que buscan estudiar un idioma en el extranjero.

Con más de 10 años de experiencia y más de 20.000 alumnos en todo el mundo, bajo la idea de que lo se vende no es simplemente un curso de idioma, sino que va mas allá, se trata de una experiencia de vida y cultura en otro país.

Además los clientes pueden también reservar alojamiento en las diferentes ciudades de destino, donde se pueden encontrar desde residencias de estudiantes o apartamentos hasta familias de acogida.

1.2 La competencia y referencias

En internet la competencia en este campo es muy elevada, y existen gran numero de webs y empresas que ofrecen los mismos servicios. Es por este motivo que es muy importante tener claro en que se quiere destacar o que se quiere conseguir con la aplicación.

Algunas de las empresas o webs que se han tomado como competencia y referencia en algunos aspectos son:

Language Bookings (<http://es.languagebookings.com/>)

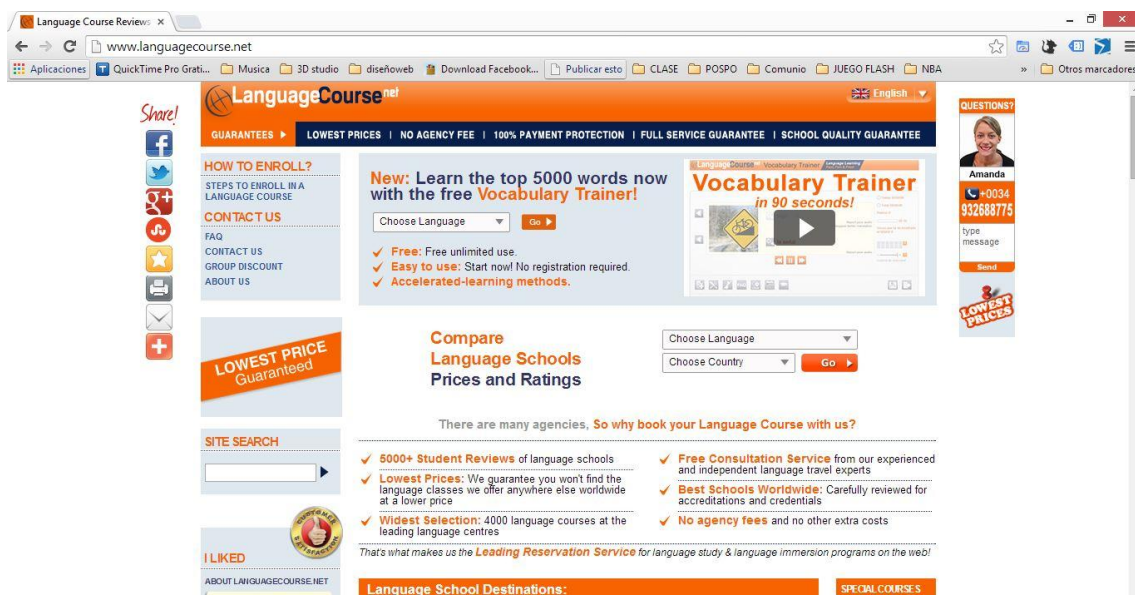


De esta empresa podemos destacar que se encuentra dentro del sector entre las mejores y con mas ventas, en cuanto al apartado técnico o de su web hay que

destacar que cuentan con un gran número de destinos y escuelas así como de idiomas en los que ofrece su página.

Nos gustaría destacar también dos aspectos interesantes, uno en cuanto a usabilidad, la utilización desde un principio de un buscador que ayuda al usuario a seleccionar no solo el idioma que quiere aprender, sino donde, durante cuánto tiempo y cuando le gustaría comenzar su curso. El segundo en cuanto a estrategia de marketing donde cuenta con un gran número de llamadas a la acción en su web (call to actions), para captar la atención del usuario y dirigirlo hacia el objetivo principal que es la venta.

Languagecourse.net (<http://www.languagecourse.net/>)



Otra de las empresas que se han considerado competencia, en este caso creemos que la estructura de su contenido esta desordenada y es un poco costoso llegar a ver todas las opciones de las que disponen.

Si se tiene que destacar algo de ellos son algunos de sus apartados donde intentan explicar al usuario como poder comprar en su web, para tratar de hacer mas fácil la navegación. También nos gustaría destacar la cantidad de medios de contacto que tienen, ya que lo que se intenta es que el cliente contacte con ellos para cualquier duda o inconveniente que pueda llegar a surgir.

En el estudio en cuanto a SEO se puede ver que esta empresa en los últimos tiempos viene realizando un buen trabajo en este sentido y ha conseguido mejorar mucho su posicionamiento.

EF idiomas

Primer referente del mercado, empresa con un gran impacto mundial. Actualmente se podría decir que son los líderes en el sector.

Su punto fuerte está en su imagen de gran marca y su cuidado diseño en todos los aspectos. También podríamos destacar aunque para ello es necesaria una alta inversión, sus campañas de publicidad destacando dentro de estas sus campañas en SEM.

Quality-Courses (web antigua)

The screenshot shows the QualityCourses website interface. At the top, there is a navigation menu with options like 'Home', 'Cursos Generales', 'Cursos Especiales', 'Cursos para Jóvenes', 'Estudia y Trabaja', and 'Contacto'. A search bar is prominently displayed with the text 'Busca tu curso de idiomas'. Below the search bar, there are filters for 'Selecciona el idioma' (set to 'Inglés') and 'Selecciona el destino' (set to 'Oxford'). There are also fields for 'Fecha de inicio' (05/08/2013) and 'Duración (semanas)' (2 Semanas). A 'Buscar' button is located below these filters.

To the right of the search bar, there is a promotional banner for 'Precio Mínimo Garantizado' featuring a smiling man. Below this, there is a section titled 'Escuelas en Oxford' which lists several course options:

Escuela	Curso General	Lecciones por semana	Duración	Niveles	Precio	Acción
Embassy English Oxford	Curso General	20	50 minutos	De principiante a avanzado	€ 591.48	Reservar
CIE Oxford (College of International Education)	Curso General	15	60 minutos	De elemental a avanzado	€ 399.28	Reservar
	Curso General	28	50 minutos	De principiante a avanzado	€ 803.52	Reservar
	Curso General	25	60 minutos	De elemental a avanzado	€ 548.08	Reservar

Below the course listings, there is a section titled 'Cursos de Inglés en Oxford' with a large image of a building in Oxford. The text below the image describes the benefits of studying in Oxford, mentioning the University of Oxford and the city's cultural and academic environment. It also mentions the availability of various services like buses and taxis.

At the bottom of the page, there is a footer with contact information, including phone numbers (+34 902 022 790 and +34 933 189 153), an email address (info@quality-courses.com), and social media links for Twitter, Facebook, and Google+. There are also logos for payment methods like VISA, MasterCard, and PayPal.

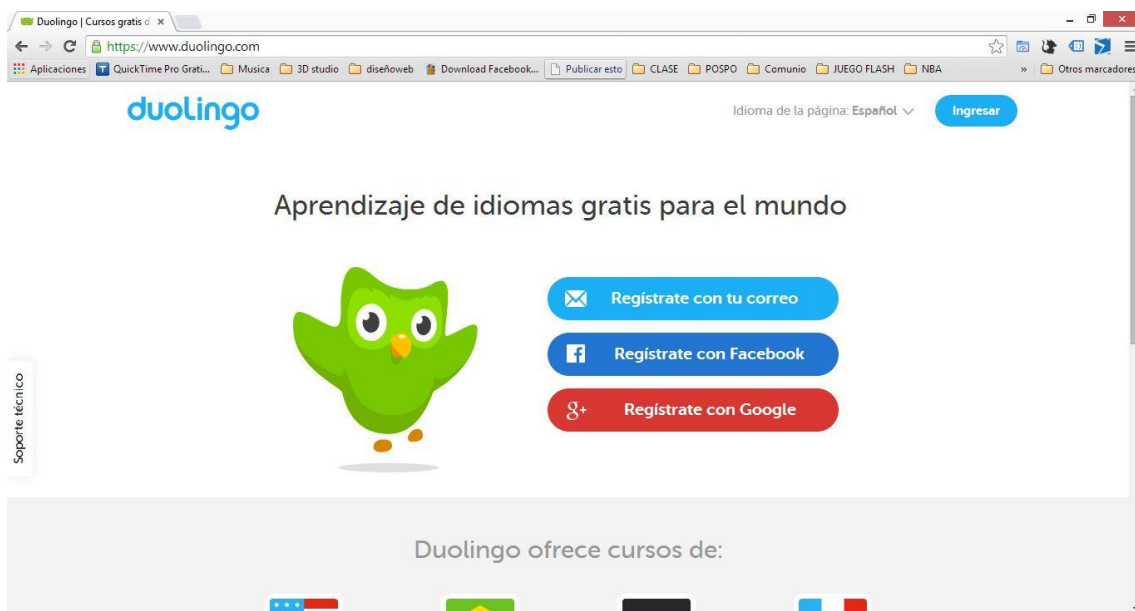
Es la web de antigua de la empresa , lanzada un año atrás y de la que se conoce a la perfección tanto los puntos positivos como negativos.

Durante este año, además de otros motivos como pudiera ser la crisis económica, se ha visto un descenso no solo en las ventas sino en el tráfico web, también ocasionado por las actualizaciones del algoritmo de google, que es la referencia en cuanto a SEO y principal buscador web mundial, y que han penalizado a la web y de las cuales se hablara mas detenidamente en el apartado de SEO.

Además como base tomamos la misma estructura de bbdd, para aprovechar la cantidad de información que ya tenemos catalogada.

En la segunda fase del proyecto cuando decidimos dar un giro y centrarnos en cambiar la web principal de la empresa y la imagen corporativa, se toma como referencia otras empresas como:

Duolingo <https://www.duolingo.com/>



De esta web cabe destacar y es una de las cosas que más nos han gustado de ellos, la sencillez y la limpieza de todo, tratan de ser claros y de que tu navegación sea simple y fácil. Con un diseño que da una visión mas moderna y actual.

Creo que cumple perfectamente con la tendencia que se está siguiendo en cuanto a diseño web en los últimos tiempos, todo muy visual, buena elección de colores, utilización de todo el ancho de la pagina...etc.

Se toma como referencia a la hora de realizar el diseño de la que será la nueva web de la empresa y de los valores que se intentan resaltar con este diseño.

2. Diseño.

2.1. Tipografía.

2.2. Color.

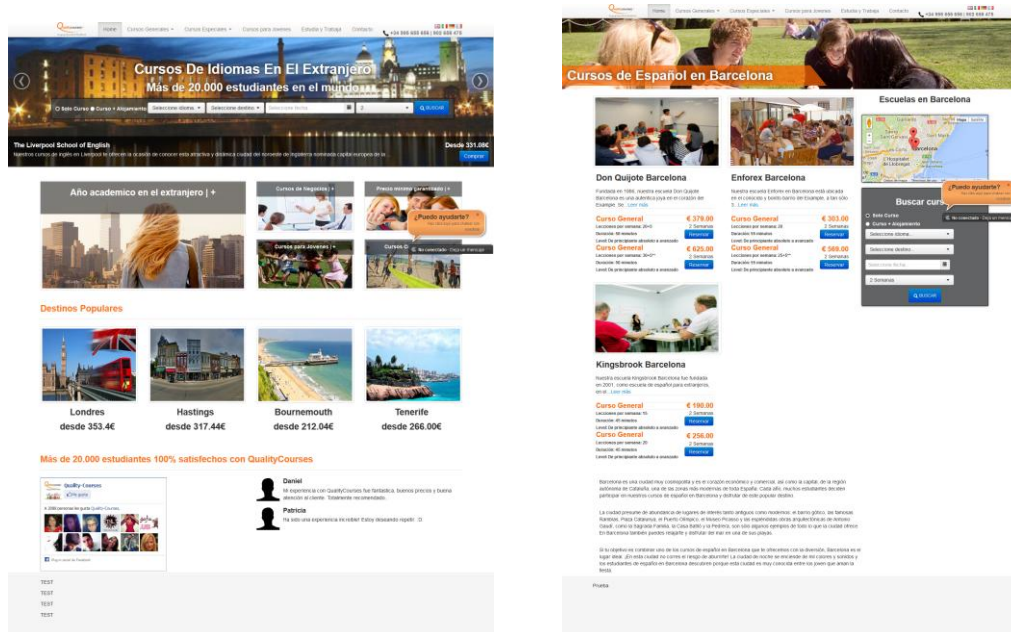
2.3. Iconos.

2.4. Maquetación.

2. Diseño

Como ya se comentó en la introducción el trabajo comenzaba con la realización de la web para los nuevos dominios para el cual se realizó un primer boceto de diseño que fue llevado luego a la versión "beta" de la web en dominios como www.aprenderidiomas.com y www.abroadlanguages.com

Aquí algunas capturas de pantalla de la versión beta publicada en los dominios antes mencionados.



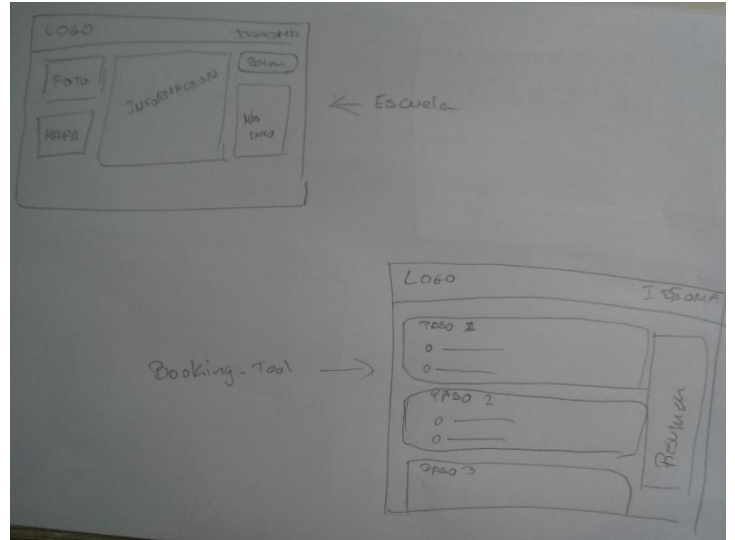
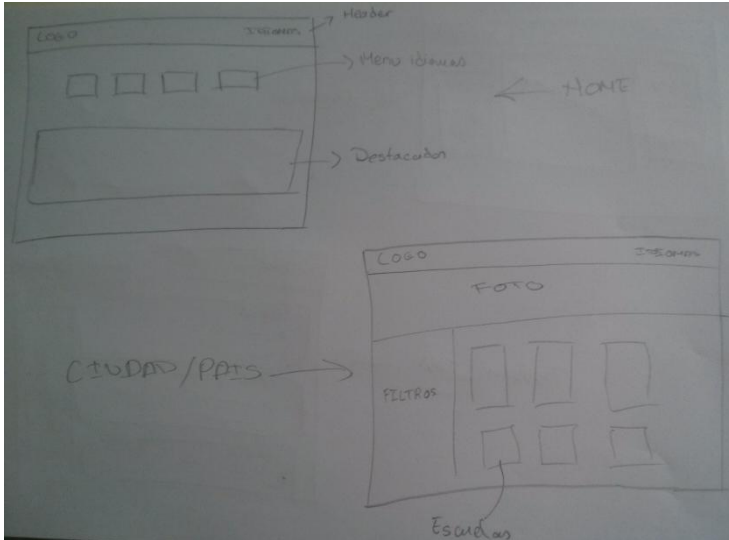
Este primer diseño original, se basaba en una Home con un buscador en la parte superior, y luego el contenido ordenado en cuadros, siguiendo la tendencia de diseño "metro". También cabe destacar que en este primer diseño se incluía un menú superior que permanecía estático al hacer *scroll* y que contaba con todas las opciones de navegación.

En la segunda fase del proyecto después de realizar una beta con estos diseños y testarlos, para lo que sería la nueva web e imagen de QualityCourses se decide hacer algunos cambios, aunque se mantienen otros en cuanto al diseño de esta beta. El porqué de estos cambios están explicados más detalladamente en el apartado de Usabilidad, ya que se refieren directamente a problemas o errores que encontramos al realizar algunos test con usuarios.

Aunque también hay otros cambios de diseño que se basan en las tendencias actuales en cuanto a diseño web y basándose también en los referentes de los cuales se han cogido ideas.

En primer lugar se realizan sobre papel unos bocetos de cómo se quiere que sea el diseño de la aplicación.

Fotos del boceto de diseño en papel.



La idea principal es mantener una web limpia y clara, fácil de identificar y entender para los usuarios. Utilizando algunas de las ideas sacadas de las webs de referencia y también de la primera versión "beta".

La realización de este nuevo diseño cuenta con varios puntos clave:

- Tipografía
- Color
- Iconos
- Maquetación

2.1 Tipografía

En primer lugar se decide la utilización de una tipografía específica, en este caso recurrimos a Google Fonts (<https://www.google.com/fonts>) porque nos da una buena cantidad de fuentes muy utilizadas en web, y una facilidad de implementación en cuanto a programación muy sencilla.

Buscamos una fuente que de una imagen moderna , con un toque juvenil y que sea optimizada para impresión, web e interfaces móviles.

La fuente seleccionada es Open Sans, esta fuente se caracteriza por las formas abiertas y una apariencia neutral, pero familiar y tiene excelentes características de legibilidad en sus letras.

Light 300

Grumpy wizards make toxic brew for the evil Queen and Jack.

Normal 400

Grumpy wizards make toxic brew for the evil Queen and Jack.

Semi-Bold 600

Grumpy wizards make toxic brew for the evil Queen and Jack.

Bold 700

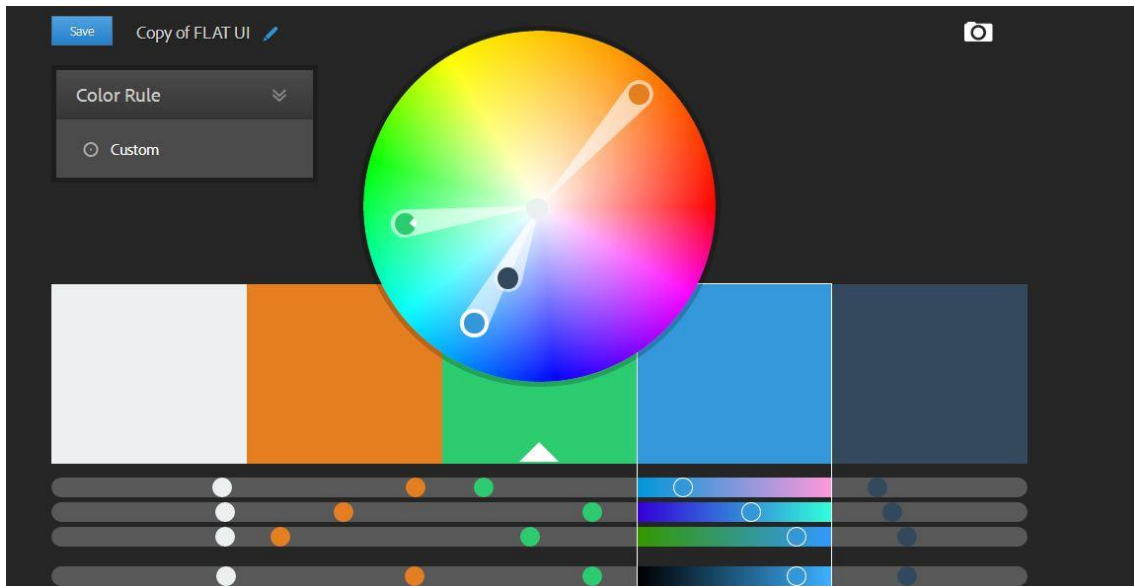
Grumpy wizards make toxic brew for the evil Queen and Jack.

2.2 Color

Como ya se ha comentado se trataba de dar una nueva imagen corporativa a la empresa, empezando por el color, donde se apuesta por cambiar el naranja que se venía usando hasta el momento, y por un azul, que sería el punto de partida de la paleta de colores de la empresa.

Con el azul se intenta conseguir una imagen más fresca y moderna, menos agresiva que el color naranja anterior, y partiendo de la base de que el azul es un color relacionado con la educación además de ser por estadísticas el color que más gusta.

Para ello partimos de la idea de utilizar un color FLAT, y una vez escogido el color principal de la marca, en este caso un azul RGB (52,152,219) o HEX(#3498DB), a partir de aquí se utiliza la web Adobe Kuler para obtener la paleta de colores que será la que a partir de este momento utilizaremos para la empresa.



Link a la paleta de colores:

<https://kuler.adobe.com/FLAT-UI-color-theme-3247888/edit/?copy=true&base=2&rule=Custom&selected=3&name=Copy%20of%20FLAT%20UI&mode=rgb&rgbvalues=0.9254901960784314,0.9411764705882353,0.9450980392156862,0.9019607843137255,0.49411764705882355,0.1333333333333333,0.1803921568627451,0.8,0.44313725490196076,0.20392156862745098,0.596078431372549,0.8588235294117647,0.20392156862745098,0.28627450980392155,0.3686274509803922&swatchOrder=0,1,2,3,4>

2.3 Iconos

Una de las cosas que se quería conseguir con esta nueva imagen era tener una consistencia visual, cosa que no se tenía en la web antigua y que tampoco se había conseguido con la primera versión beta.

Se decide la utilización de iconos que refuerzan la comunicación de lo que la empresa quiere transmitir además de dar una imagen más moderna y actual.

Y para ello se necesitaban unos iconos acordes a la nueva imagen que quería transmitir y consistentes visualmente entre ellos.

Se recurre a la web iconfinder.com donde adquirimos una serie de iconos adaptados al contenido que vamos a tener en la web.

Algunos de los iconos obtenidos son los siguientes:



2.4. Maquetación

Con las principales directrices del diseño definidas, llegamos al momento de la maquetación, y la transformación de la idea, colores y formas que tenemos en diseño, en código HTML 5 y CSS.

Para ello y con la premisa de que queremos que nuestra web sea responsive y se adapte a todas las pantallas de los dispositivos actuales se decide utilizar el framework de HTML y CSS Bootstrap.



Bootstrap es una serie de herramientas predefinidas en HTML , CSS y JavaScript que nos permiten en primer lugar ahorrar mucho tiempo a la hora de teclear código, pero sobretodo y la mayor ventaja que ofrece es la compatibilidad con la mayoría de navegadores principales. Esto ahorra muchos problemas y tiempo. Además de que también cuenta con una gran comunidad de desarrollo que nos permite salir casi de cualquier problema.

Básicamente lo que bootstrap hace por nosotros es darnos una base CSS y JS preparado con un gran número de clases que esta optimizado también para un mejor tiempo de carga de la pagina y un código limpio.

Las opciones de las que cuenta son inmensas, desde la creación de divs con clases para cada tipo de dispositivo, formularios, botones, efectos, sliders...etc

Este framework es utilizado hoy en día por muchísimas empresas de gran envergadura, comenzando por Twitter que son sus creadores.

.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1
.col-md-8								.col-md-4			
.col-md-4				.col-md-4				.col-md-4			
.col-md-6						.col-md-6					

Para su utilización lo primero que se tiene que tener claro es que bootstrap divide el 100% del ancho de pantalla en 12, es decir para utilizar el 100% tenemos que añadir las clases CSS de 12, y a partir de ahí dividimos en columnas como se quiera. Esto funciona de la misma manera en todos los dispositivos.

La estructura de la web se basa en 6 pantallas en las cuales se cargará el contenido dinámicamente dependiendo de la url:

- Home
- Ciudad
- País
- Escuela
- Formulario de inscripción
- Contacto
- Ofertas

Y un diseño que se utiliza en otras por defecto, y que se puede utilizar para generar nuevo contenido como:

- Pagina de Gracias
- Cursos de larga duración
- Trabajar y estudiar

Algunas capturas del código HTML

Menú bootstrap

```
<div class="navbar navbar-default navbar-fixed-top">
  <div class="col-md-12">
    <div class="navbar-header">
      <a href="{home_url}" class="navbar-brand"></a>
      <button class="navbar-toggle" type="button" data-toggle="collapse" data-target="#navbar-main">
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
    </div>
    <div class="navbar-collapse collapse" id="navbar-main">
      <ul class="nav navbar-nav navbar-right">
        <li class="dropdown language"><a style="font-weight: 300;" class="dropdown-toggle" data-toggle="dropdown" href="#">{menu_idioma}: <span style="font-weight: 300;" class="dropdown-menu">
          <ul class="dropdown-menu">
            <li>{lang_engine}</li>
          </ul>
        </li>
        <li><a href="{contact_url}" target="" style=""> {menu_contact}
        <li><a href="{contact_url}" target="" style=""> +34 933 189 1
      </ul>
    </div>
  </div>
</div>
```

Clases de los div de bootstrap:

```
<div class="col-lg-4 col-sm-4 col-xs-4 col-md-4 press-review ">
```

Ventana modal

```
<!-- Modal -->
<div class="modal fade" id="myModal" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">
  <div class="modal-dialog">
    <div id="map_canvas" class="modal-content col-md-12" style="height: 480px; ">
      <div id="map_canvas">
      </div>
    </div>
  </div>
</div>
<!-- Modal -->
```

Llamadas "xtemplate"

```
{nombre_ciudad}
```

Resultado final

Home



Language of Page: English

Contact

+34) 933 189 153 / 902 022 790

Language courses abroad

What language would you like to learn?



English



Spanish



Italian



German



French



Portuguese



LEARN

Find the language course that suits you.



LOWEST PRICE

Minimum price guaranteed.



DISCOVER

Live an incredible experience abroad.

Online - May I help you?

Home (2)



Language of Page: English

Contact

+34) 933 189 153 / 902 022 790

LEARN

Find the language course that suits you.

LOWEST PRICE

Minimum price guaranteed.

DISCOVER

Live an incredible experience abroad.



Juniors Courses



Work and Study



Business Courses



Long stay courses

Recommended language schools



2 weeks from 419 €

ILAC Toronto

★★★★★

Best value for money.

Book from: 104.75€

More details



2 weeks from 737 €

Embassy London

★★★★★

Best value for money.

Book from: 221.10€

More details



2 weeks from 415 €

Zoni Language Miami

★★★★★

Best value for money.

Book from: 124.50€

More details



2 weeks from 357 €

Campus Internacional Hispania

★★★★★

Best value for money.

Book from: 107.10€

More details

Online - May I help you?

Navegación

The screenshot shows the Qualitycourses website navigation menu. A modal window titled "Choose a destination" is open, displaying a list of countries: England, Canada, United States, Scotland, Ireland, Malta, Australia, New Zealand, South Africa, and Cyprus. The background shows the website header with the Qualitycourses logo, a contact number (+34) 933 189 153 / 902 022 790, and a language selector set to English. Below the navigation menu, there are three main sections: LEARN (Find the language course that suits you.), LOWEST PRICE (Minimum price guaranteed.), and DISCOVER (Live an incredible experience abroad.).

Ciudad / País

The screenshot shows the Qualitycourses website displaying English courses in the United States. The header includes the Qualitycourses logo, a language selector set to English, and a contact number (+34) 933 189 153. The main banner features a cityscape and the text "English courses in United States". Below the banner, there are three main sections: LEARN (Find the language course that suits you.), LOWEST PRICE (Minimum price guaranteed.), and DISCOVER (Live an incredible experience abroad.).

Below the banner, there is a search filter for "Find your course in United States" with options for "Course only" and "Course + Accommodation". The search results show three course listings:

Course Name	Location	Duration	Price	Lessons per week
Zoni Language Miami	Miami	2 weeks	from 415 €	15
The Language Academy	Fort Lauderdale	2 weeks	from 434 €	20
ALCC New York	New York	2 weeks	from 451 €	18

Escuela



France Langu Opera

27 Boulevard des Italiens - 75002 Paris - France

★★★★★
5/5 (15)

Book now

General information and availability

This France Langu located in the elegant and lively area of Paris, Opera, is particularly famous for its splendid architecture, trendy fashion shops, and numerous bars and restaurants, offering many opportunities to live the Parisian life. The school is easy accessible via metro or bus, making it easy for visitors to explore their surroundings and enjoy the many cultural activities that one of the best student cities in the world has to offer.

The facility itself is in a typically Parisian building on the Boulevard des Italiens, away from the bustle of the street. The school is Spacious, elegant with pleasant premises, the 12 classrooms spread over 2 floors offer a small learning environment. For leisure and relaxation the school is equipped with a student lounge, drinks and snacks vending machines, an internet area with computers, and free WIFI access.

Facilities

-
-
-
-



- Minimum price guaranteed
- 7% discount
- Free online course
- Registration fee included
- No additional fees

[- Click here to chat](#)

Booking Tool (Escuela)

Book your course now

1° Start day & lesson duration

Starting date:

Duration (weeks):

2° Select course

Courses	Total price	Book from	Duration (weeks)
<input checked="" type="radio"/> General Course 25 lessons per week +info	260.00€	52.00€	2 weeks

3° Select accommodation

Accommodation type	Students per room	Board	Price
<input type="radio"/> I don't need accommodation			0€
<input checked="" type="radio"/> Homestay, Single Room	1	Breakfast	400.00€
<input type="radio"/> Homestay, Single Room	1	Half Board	420.00€

Booking summary

Course
General Course 260.00€

Accommodation
Homestay, Single Room 400.00€

Total: 660.00€

Payable now: 52.00€

To book this course you only have to pay this small deposit, which will be deducted from the total price.

Balance: 608.00€

Book now

Online - May I help you?

Formulario de inscripción

You're almost there! You're just a minute away from completing your booking...

Student	Booking summary
<p>First name: <input type="text" value="First name"/></p> <p>Last name: <input type="text" value="Last name"/></p> <p>Birthdate: <input type="text"/></p> <p>Email: <input type="text" value="Email"/></p> <p>Confirm email: <input type="text" value="Confirm email"/></p> <p>Nationality: <input type="text"/></p> <p>Passport/ID: <input type="text"/></p> <p>Telephone: <input type="text"/></p> <p>Language Level: <input type="text"/></p> <p>Gender: <input type="radio"/> Male <input type="radio"/> Female</p> <p><input type="checkbox"/> I would like to take out your cancellation, health and travel insurance. 56.00 €</p> <p><input type="checkbox"/> I would like to receive your newsletter.</p> <p>Comments: <input type="text"/></p> <p><input type="checkbox"/> I have read and accept the Terms and Conditions.</p>	<p>Embassy Cambridge General Course 20 lessons per week 719€</p> <p>Accommodation Homestay, Single Room 469€</p> <p>Arrival: 22/06/2014 Departure: 05/07/2014</p> <p>Total: 1188€</p> <p>Payable now: 215.7€</p> <p>To book this course you only have to pay this small deposit, which will be deducted from the total price.</p> <p>Balance: 972.3€</p> <p>May I help you? Click here to chat with us</p> <p>Offline - Leave a message</p>

Formulario de inscripción (2)

Payment Type	Information
<p><input type="checkbox"/> I have read and accept the Terms and Conditions.</p> <p><input checked="" type="radio"/> Credit card </p> <p><input type="radio"/> Bank transfer </p> <p></p> <p><input type="button" value="Pay now"/></p>	<p>Balance: 972.3€</p> <p>Security and reliability</p> <p>When booking with QualityCourses.com, you will not only get the best value and market price; you are also guaranteed to obtain a competent and powerful ally that will stand up for your best interests with the schools or providers in the unlikely case something unexpected happens.</p> <p>Booking 100% guaranteed</p> <p>There is extremely slim chance of something going amiss once your spot has been confirmed by the school or service provider. In fact, we are so certain of our management that if, by any reason, you do not hold your vacancy at the school; we will reimburse all your deposit fees and find you the most suitable option available.</p> <p>Personal information</p> <p>Please check that all of your personal information is correct before submitting it. We will use this information to book your course and ensure that the email address you provide is correct so you will be receiving all the information you need for your experience abroad.</p> <p>May I help you? Click here to chat with us</p> <p>Offline - Leave a message</p>

Paginas gracias

Qualitycourses

Language of Page: English ▾

✉ contact

☎ (+34) 933 189 153 / 902 022 790

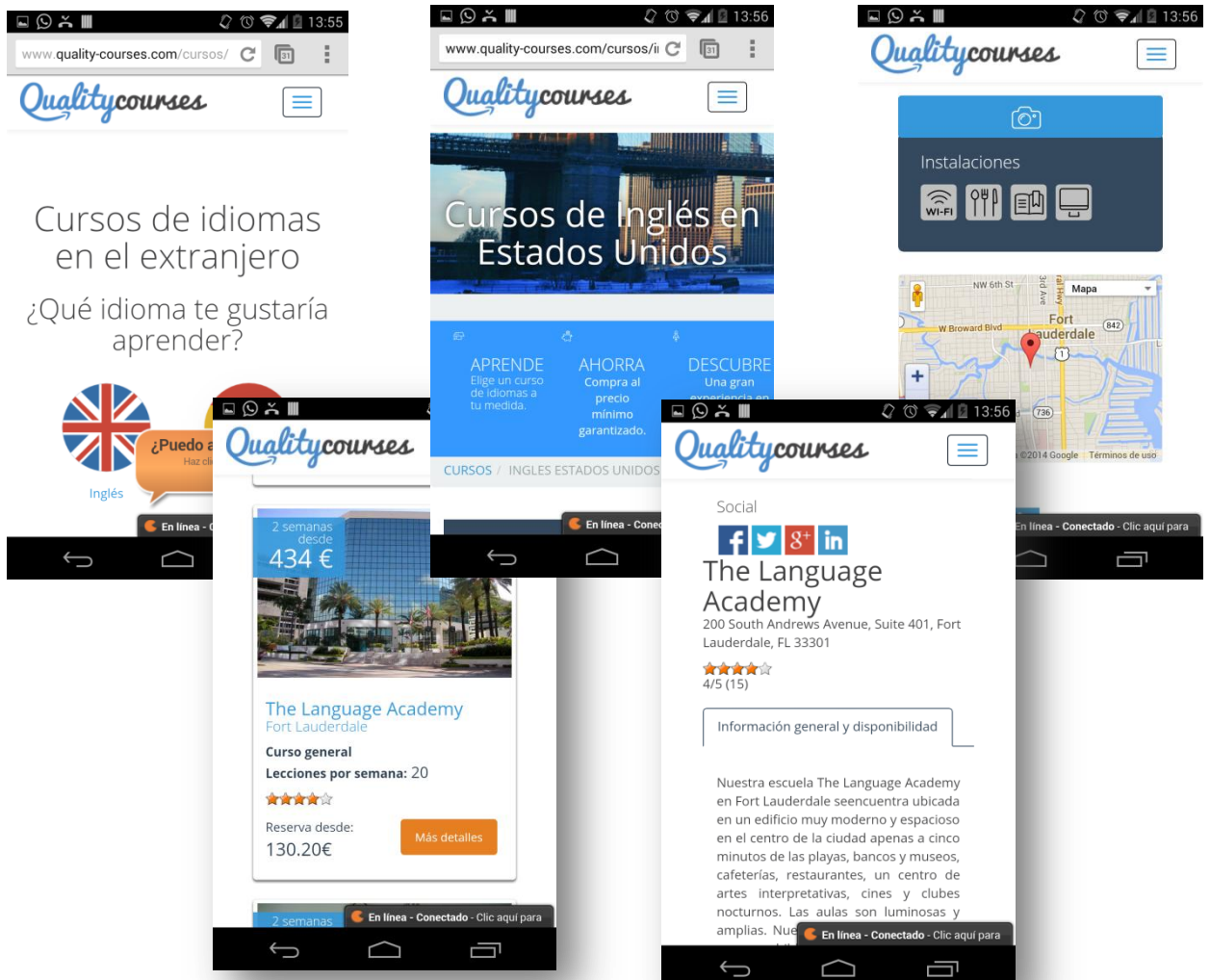
¡Pack your bags!

Thank you for booking with QualityCourses! In the coming hours you will receive an email on your booking.

Should you have any further needing please you can contact us by phone (from Monday to Friday) or email to info@quality-courses.com.

Keep in touch!

Responsive



3. Base de Datos.

4. Programación backend.

4.1. El núcleo.

4.2. Módulos.

4.2.1. Url Engine

4.2.2. Lang Engine

4.2.3. Lang Menu

4.2.4. Meta tags Module

4.2.5. Footer

4.2.6. Booking tool

4.3. Home

4.4. Ciudad/País

4.5. Escuela

4.6. Formulario de inscripción

4.7. Checkout.php y ok.php

4.8. Contacto

4.9. Ofertas

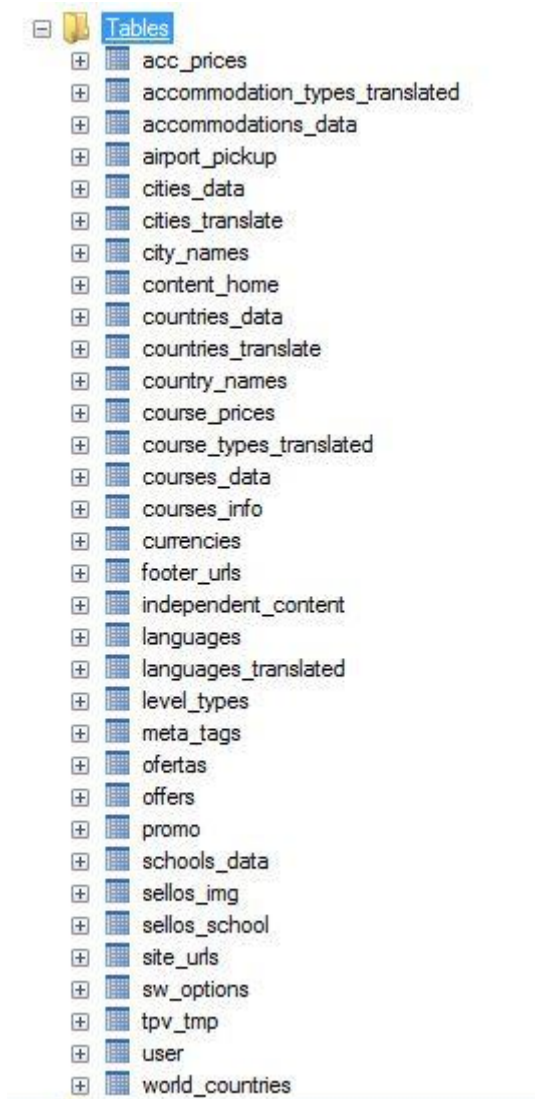
4.10. Paginas default

4.11. Ajax/ Jquery

3. Base de Datos

Como se ha comentado para la base de datos se intenta aprovechar al máximo la base de datos de la anterior web de QualityCourses, ya que es una base de datos bien estructurada y además no ahorra mucho tiempo.

Aunque será necesario hacer pequeños cambios, nuevos atributos en alguna de las tablas, e incluso alguna nueva tabla necesaria para almacenar el nuevo contenido de la web.



Estas son las tablas que se utilizan en la web, la mayoría de ellas son rescatadas de la anterior versión de la aplicación.

4. Programación backend

Este apartado supone la parte más importante del proyecto. Se trata de la programación backend, realizada en PHP, Javascript y JQuery y apoyada sobre la base de datos en MySQL.

4.1 El núcleo

Hablando de la base o núcleo de la web hay que nombrar las herramientas principales sobre las que se apoya todo el proyecto, se usa una serie de herramientas para programadores disponibles en la web phptoolcase.com

Cuando se habla del núcleo, se habla de la parte de la programación que hace que todo funcione, y para hablar de ello hay que detenerse a explicar las herramientas que se utilizan.

¿Qué es PHP Tool Case?

Php tool case es un proyecto para acelerar el proceso de programación en php, mediante diversas herramientas que puedan reducir significativamente las líneas de código.

Entre las herramientas que ofrece *Php Tool Case* en este proyecto se han utilizado las siguientes:

- Autoloader y extensión simplewebsite. (PtcHm)
- Debugger & Logger (PtcDebug).
- MySql Helper (PtcDb).

Además de estas clases, también se utiliza la clase *Xtemplate*, que nos facilita la separación del código *HTML* y el *PHP*. Permitiéndonos entre otras cosas una mayor facilidad para la carga de templates.

La base de la web se apoya en la clase Autoloader, esta herramienta permite la carga automática de clases ahorrando tiempo en comparación a la inclusión manual de las clases.

Funciona cargando todas las clases desde un directorio dado. A parte también se utiliza una extensión del *Autoloader* llamada *SimpleWebsite*.

Esta extensión básicamente lo que hace es la creación de un objeto *\$sw* donde se almacenan todos los datos importantes y más usados en todo el código del website.

Para explicarlo un poco mejor se pasa a explicar algunos de los archivos más importantes y donde se configura todo el funcionamiento del código.

En primer lugar el archivo sw-config.php, básicamente en este archivo lo que se hace es configurar todas las rutas que se utilizarán en la web, además de la carga del Autoloader y de añadir a este el directorio desde el cual cargará las demás clases.

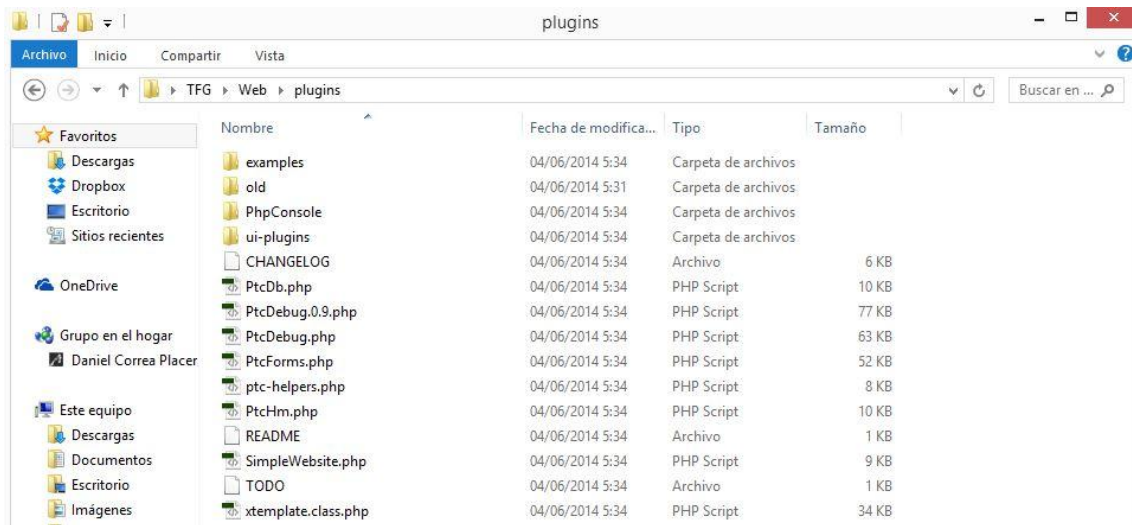
```
1 <?php
2 # short name for DIRECTORY_SEPARATOR
3     define('SEP', DIRECTORY_SEPARATOR);
4
5 # main path of application
6     define('MAIN_PATH', realpath(dirname(__FILE__)).SEP);
7
8 # content path of application
9     define('CONTENT_PATH', 'content'.SEP);
10
11 # includes path of application
12     define('INCLUDES_PATH', 'plugins'.SEP);
13
14 # modules path of application
15     define('MODULES_PATH', 'modules'.SEP);
16
17
18
19
20 # include the phptoolcase handyman(classes autoloader)
21     require_once('./plugins/PtcHm.php');
22
23 # add path for core includes
24     PtcHandyMan::AddPath('./plugins/');
25
26 ?>
```

Su explicación es sencilla, en las primeras líneas de código hasta la 15 definimos una variable constante para cada uno de los directorios que luego utilizaremos en todo el código de la web. Esto se hace aunque después en la realización del código es decisión nuestra si queremos utilizar las constantes para establecer las rutas o directamente escribimos la ruta completa.

Más que nada sirve para ahorrarnos tiempo en la escritura de las rutas de los archivos que componen el site.

La parte más importante de este archivo son las líneas 21 y 24, en la primera de estas cargamos el archivo PtcHm.php que es el que incluye la clase Autoloader y en la segunda utilizamos ya esta clase para añadirle el directorio desde el cual queremos que cargue las clases que vamos a incluir.

Es en este directorio donde tendremos las demás clases que vamos a utilizar y que ya he comentado antes (MySQL helper, Debugger & Logger, Xtemplates)



Otro de los archivos de configuración importantes para que el sistema funcione correctamente es el connect.php

Una vez creado el objeto \$sw , nos encargamos de establecer los datos de conexión a la db.

```
1 <?php
2 if(isset($sw))
3 {
4     $swTblPrefix="sw_";
5
6     $swTable['options']=$swTblPrefix."options";
7
8
9     $db['host']='localhost';
10    $db['user']='d150210_web';
11    $db['pass']='qc4326';
12    $db['db_name']='d150210_qc_ng_web';
13    $db['db_encoding']='UTF8';
14
15    $sw->addVars($db, 'db');
16
17
18    # add database query class (from PhpToolCase)
19    $sw->addTool('PtcDb');
20
21    # connect to database
22    $sw->dbConnect($sw->db['host'],$sw->db['user'],$sw->db['pass'],$sw->db['db_name'],$sw->db['db_encoding']);
23    # get application settings
24    $sw->getSettings();
25 }
26 <?>
```

Su funcionamiento es sencillo, primero nos aseguramos de que tenemos el objeto \$sw creado con un if(), a partir de aquí lo primero que se hace es crear dos variables \$swTblPrefix (que incluye un prefijo que añadimos a las tablas de la db que cuentan con información necesaria para el funcionamiento) y \$swTable['options'] (que contiene el nombre de la tabla que tendrá una serie de opciones que son necesarias para el funcionamiento). Estas variables son tomadas

más tarde por la clase SimpleWebsite que se conectará a la db para obtener los datos de "options" y también los guardara en el objeto \$sw.

En las líneas 9 hasta la 13 se crea el array \$db con los datos de conexión a la base de datos, y los añado al objeto principal \$sw en la línea 15 (\$sw->addVars(\$db,'db'))

El siguiente paso es añadir la herramienta PtcDb que está cargada por el Autoloader al objeto \$sw (\$sw->addTool('PtcDb')).

Esto nos permite utilizar las funciones que la herramienta MySql helper nos propone directamente sobre el objeto \$sw. Conseguimos que a partir de ahora toda la utilización que demos a la base de datos quede sobre el objeto \$sw.

Con esto ya solo queda que utilizando las forma que nos da esta herramienta realizar la conexión a la db con los parámetros antes guardados y hacer que se ejecute la conexión.

```
$sw->dbConnect($sw->db['host'],$sw->db['user'],$sw->db['pass'],$sw->db['db_name'],$sw->db['db_encoding']);
```

```
$sw->getSettings();
```

En este punto esta toda la configuración del núcleo de la web establecida y podemos pasar al archivo encargado de unir todo (index.php).

```
1 <?php
2 session_start();
3 error_reporting(E_ERROR);
4
5
6 //Incluimos archivo config
7 include_once('./sw-config.php');
8
9
10 require_once(MAIN_PATH.INCLUDES_PATH.'PtcDebug.php');
11 # replace php built in error handler and continue execution if fatal error occurs
12
13 $debug_options=array
14 (
15     'url_key'           =>     'debug',
16     'url_pass'         =>     'true',# use a complicated string!
17     'replace_error_handler' =>     false, # has been replaced already
18     //'allowed_ips'     =>     array(1.2.3.4,2.3.4.5),
19     'show_interface'   =>     true
20 );
21 PtcDebug::load($debug_options);
22
23 //inicializamos la class simplewebsite
24 $sw=new SimpleWebsite('',true);
25 $sw->magicQuotes();
26
27 //Conexion a la db y web options
28 require_once(MAIN_PATH.'connect.php');
29 //Include los plugins/modules
30 require_once(MAIN_PATH.'sw-plugins.php');
31
32 log_msg($sw->coreOptions['base_dir'],'BASE DIR');
33
34 if(!$sw->pageExists){ $sw->checkUri(); } # check if url exists in db
35 if($sw->pageExists){ $sw->SWIplEngine(); }
36
37 ?>
```

Antes de comentar nada en este archivo cabe destacar que tenemos que configurar en el servidor el archivo `.htaccess` para que todas las url abran este archivo.

Lo primero que hacemos es iniciar la session y mediante la función (`error_reporting(E_ERROR)`) establecer que solo se nos muestre en pantalla los errores graves, *esta función es solo para la fase de ejecución*, para evitar que se muestren posibles Warnings que no influyen en la ejecución correcta de la aplicación. *(En fase de desarrollo es aconsejable no ejecutar este código para que nos muestre todos los errores)*.

Incluimos el archivo `config.php` seguidamente. Y también incluimos la clase Debugger & Logger, que podría ser incluida automáticamente por el Autoloader pero que como cualquier otra podemos incluir manualmente, para asegurarnos de su carga en el momento correcto.

Esta herramienta nos parece la más interesante de las que podemos encontrar en Php Tool Case nos da una interfaz visual donde podemos controlar todo lo que está pasando en nuestra aplicación, entre otras cosas:

- *Saber el contenido de cualquier objeto.*

- *Ver el contenido de cualquier variable(Local, Global, de sesión, array...etc) en el momento en que queramos saberlo.*

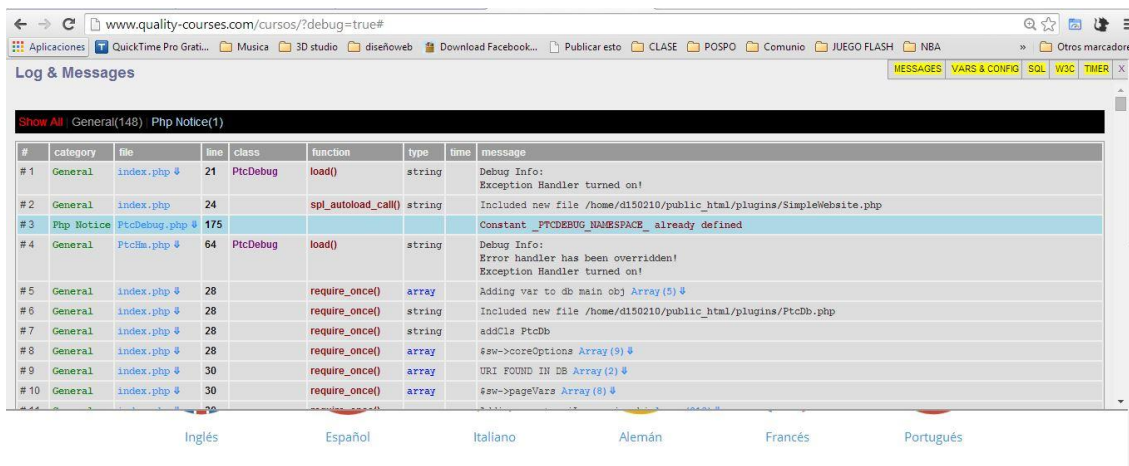
- *Debido a su implementación con la clase Ptcdb nos permite ver todas las consultas que se están ejecutando en el momento y ver también el tiempo de ejecución de esta consulta.*

- *Observar el orden de todos los includes de las clases que se añaden a la aplicación.*

- *Tiempo global de ejecución.*

- *Muestra errores y notificaciones.*

Se trata de una gran ayuda en la producción además de que nos muestra en cada caso el archivo y la línea en que está ocurriendo cada notificación. Pudiendo incluso abrir una visualización del código que la está causando.



Para la configuración de esta herramienta podemos utilizar algunas opciones que definimos en antes de ejecutarlo.

```

$debug_options=array
(
    'url_key'           => 'debug',
    'url_pass'         => 'true', # use a complicated string!
    'replace_error_handler' => false, # has been replaced already
    //'allowed_ips'    => array(1.2.3.4,2.3.4.5),
    'show_interface'  => true
);
PtcDebug::load($debug_options);

```

Cada una de las opciones que añadimos al array \$debug_options tiene una función, entre ellas las más importantes son "url_key" y "url_pass" que nos define las variables que tendremos que poner en la url para ejecutar el Debugger , "replace_error_handler" que reemplaza el controlador de errores por defecto de php para que estos sean mostrados en la consola de mensajes del Debugger y el "show-interface" que nos permite activar la interfaz grafica.

Con la línea (PtcDebug::load(\$debug_options)) cargamos las opciones definidas en el debugger, si no establecemos opciones, tiene unas por defecto.

Lo siguiente es crear el objeto \$sw inicializando la clase SimpleWebsite que ha sido cargada por el Autoloader en el archivo config.php y ejecutamos la función magicQuotes() incluida en la misma clase, para activar el uso de comillas de escape.

Ahora que ya tenemos el objeto \$sw incluimos el archivo connect.php y también el archivo sw-plugins.php que será el siguiente archivo que comentaremos.

Por último ejecutamos la función (checkUri();) que está incluida en el archivo sw-plugins.php y que nos controla la existencia de la url y cuando obtenemos su respuesta ejecutamos (SWTplEngine();) que será encargado de cargar el archivo

requerido para esa url. *(Estas funciones serán explicadas más detalladamente a continuación)*

Como detalle hay que fijarse en la línea 32 de código:

```
"log_msg($sw->coreOptions['base_dir'],'BASE DIR');"
```

Una estructura que a partir de ahora veremos mucho en algunas partes del código, esta sentencia nos permite enviar al Debugger un mensaje o una variable en concreto y nos permite que podamos ver su valor en ese instante. En este caso queremos ver el valor de una propiedad del objeto \$sw , concretamente el coreOptions['base_dir'] y a la que le damos un nombre para identificarlo en la interfaz del Debugger "BASE DIR", esto nos dará el valor de esa propiedad en ese momento exacto, y esta propiedad en concreto es la que recoge de la base de datos en la tabla options que habíamos comentado antes y que cuenta con algunas opciones para el funcionamiento de la web.

En el archivo sw-plugins.php cargamos los módulos o plugins que serán utilizados en todas las páginas de la aplicación. Después habrán más módulos que serán cargados a medida de que sean necesarios.

En este caso son tres:

```
1  <?php
2  require_once('./modules/url_engine/url_engine.php');
3
4  require_once('./modules/lang_engine.php');
5
6  require_once('./modules/lang_menu/lang_menu.php');
7
8  ?>
```

El primer modulo que cargamos es url_engine.php, que es donde se encuentra la función (checkUri();) comentada anteriormente y que controla si la url a la que estamos intentando acceder existe y si es el caso y dependiendo de que url sea establece el archivo y la plantilla que tiene que cargar.

El segundo modulo es lang_engine.php que es el encargado de controlar el lenguaje de la pagina y el archivo de diccionario de lenguaje que tiene que cargar.

Y en tercer lugar cargamos el modulo lang_menu.php que se trata del encargado de crear el menú de idiomas de la pagina que nos permitirá en cualquier url cambiar de un idioma a otro sin cambiar de url.

4.2 Módulos

4.2.1 Url Engine

Este modulo es el encargado de comprobar la url y decidir que tiene que cargar si la url existe en nuestra base de datos dependiendo del tipo que le hayamos definido. Hay varios tipos y podemos añadir nuevos, los más importantes: home,city,country, school...etc

El tipo de pagina nos indicara que plantilla y que archivo tenemos que cargar.

```
if(empty($uri)){ $uri=parse_url($_SERVER['REQUEST_URI']); }
else if(!is_array($uri) || (!array_key_exists('path',$uri))){ $uri['path']=$uri; }
$sw->requestUri=$uri;
$sw->getSettings();

$sw->requestUri['path']=str_replace($sw->coreOptions['base_dir'],"/",$sw->requestUri['path']);

...
$sw->pageVars=$sw->readRow("site_urls",array("full_uri"=>$sw->requestUri['path'], "status"=>1));
```

En primer lugar recogemos la url y la guardamos en la variable \$uri si esta no existe y además la añadimos a una variable de nuestro objeto \$sw. Con esta variable y la función *str_replace* lo que hacemos ahora es dividir la url separando lo que sería la url base, que ya la teníamos en una variable del objeto , y la parte que le sigue a partir del símbolo "/".

Cuando tenemos ya el path de nuestra url lo buscamos en la tabla "site_urls" de la base de datos. Con esto obtendremos un resultado si existe y el resultado lo guardamos en la variable *pageVars* dentro del objeto \$sw.

```
switch($sw->pageVars['type'])
{
    case "school":
        if(!$sw->pageVars['tpl_page']){ $sw->pageVars['tpl_page']="school.xtpl"; }
        $sw->coreOptions['tpl_main']="school-tpl.php";
        $sw->addVars($school_data=$sw->readRow("schools_data","id:".$sw->pageVars['ref']),"schoolData");
        log_msg($school_data,'the school data!');
        break;
    case "becascountry":
        $sw->pageVars['tpl_page']="becas.xtpl";
        $sw->coreOptions['tpl_main']="becas-tpl.php";
        break;
    case "becascity":
        $sw->pageVars['tpl_page']="becas.xtpl";
        $sw->coreOptions['tpl_main']="becascity-tpl.php";
        break;
    case "city":
        $sw->pageVars['tpl_page']="resultados.xtpl";
        $sw->coreOptions['tpl_main']="city-tpl.php";
        if(!$sw->pageVars['tpl_page']){ $sw->pageVars['tpl_page']="resultados.xtpl"; }
        $sw->addVars($city_data=$sw->readRow("cities_data","id:".$sw->pageVars['ref']),"cityData");
        break;
    case "country":
        if(!$sw->pageVars['tpl_page']){ $sw->pageVars['tpl_page']="resultados.xtpl"; }
        $sw->coreOptions['tpl_main']="country-tpl.php";
        break;
    case "home":
        $sw->pageVars['tpl_page']="home.xtpl";
        $sw->coreOptions['tpl_main']="home-tpl.php";
```

Ahora que tenemos los datos obtenidos de consulta a la base de datos si existe la url, y montamos un switch con el *type* de la url, dependiendo del *type* como habíamos dicho antes se cargara un documento y una plantilla.

También en algunos casos si es necesario para agilizar la carga podremos guardar una variable dentro del objeto \$sw con datos que nos puedan hacer falta dependiendo de la pagina a la que nos dirijamos. Como por ejemplo en el caso de school, donde añadimos una variable al objeto \$sw con los datos de las escuelas, obtenidos de la base de datos.

Otro de los campos que obtenemos de la url es la *ref*, que se trata de la referencia hacia la que apunta esa url, por ejemplo si es una escuela, seria la id de la base de datos donde se encuentran los datos de la escuela en concreto.

Se añade un "case" por cada uno de los tipos de url que tenemos, indicándole si es el caso cual será la plantilla a utilizar (*tpl_page*) y cuál será el archivo php que se encargara de la gestión de ese tipo (*tpl_main*).

```
break;
case "home":
    $sw->pageVars['tpl_page']="home.xtpl";
    $sw->coreOptions['tpl_main']="home-tpl.php";

break;
case "inscription-form":
    $sw->pageVars['tpl_page']="inscription.xtpl";
    $sw->coreOptions['tpl_main']="inscription-tpl.php";
break;
case "thanks":
    $sw->pageVars['tpl_page']="thanks.xtpl";
    $sw->coreOptions['tpl_main']="thanks-tpl.php";

break;
case "contact":
    $sw->pageVars['tpl_page']="contact.xtpl";
    $sw->coreOptions['tpl_main']="contact-tpl.php";
break;
case "work":
    $sw->pageVars['tpl_page']="work.xtpl";
    $sw->coreOptions['tpl_main']="work-tpl.php";
break;
case "long":
    $sw->pageVars['tpl_page']="work.xtpl";
    $sw->coreOptions['tpl_main']="work-tpl.php";
break;
case "business":
    $sw->pageVars['tpl_page']="work.xtpl";
    $sw->coreOptions['tpl_main']="work-tpl.php";
break;
```

También establecemos una plantilla y un archivo php por defecto, para evitarnos posibles tipos que no estén configurados o para utilizarlo en las paginas donde la plantilla o el contenido sea igual.

Con esto podremos generar nuevas páginas de manera fácil.

```

        default:
            $sw->pageVars['tpl_page']="default.tpl";
            $sw->coreOptions['tpl_main']="default-tpl.php";
        }
    }
    if($sw->pageVars['tpl_main']){ $sw->coreOptions['tpl_main']=$sw->pageVars['tpl_main']; }
}

if(!$sw->pageVars){ $sw->pageVars=$sw->readRow($swTable['pages'],array("full_uri"=>$sw->requestUri['path',"status"=>1)); }

if($sw->pageVars)
{
    $sw->pageExists=true;
    log_msg($sw->requestUri,'URI FOUND IN DB');
    log_msg($sw->pageVars,$sw->pageVars);

    $_SESSION['ui_lang']=$sw->goFast('languages','id',$sw->pageVars['lang_id'],'lang');
    $_SESSION['ui_lang_id']=$sw->pageVars['lang_id'];
}
else
{
    //echo "no existe 404";
    header("HTTP/1.0 404 Not Found");
    header('Location: http://www.quality-courses.com');
    /*$sw->pageExists=false;
    $sw->pageVars['tpl_page']='404.tpl';
    log_msg($sw->requestUri,'URI NOT FOUND IN DB'); */
}
}

```

Por último si la url existe y el tipo es correcto, además de guardar una variable dentro del objeto donde indicamos que la url es correcta, creamos las primeras variables de session y que serán los indicadores del idioma. (También recogidos de la consulta en la base de datos de la url)

Depende de la url obtendremos desde la base de datos que es el lang_id que se trata de la id del idioma, los idiomas están definidos en la tabla de *languages*, y de esta manera mediante una consulta a la base de datos de nuevo creamos las variables \$_Session['ui_lang'] y \$_Session['ui_lang_id'], el primero con las iniciales del idioma al que corresponde a esa url y en segundo lugar la id del idioma.

En caso de que la url no exista, indicamos un error 404, mediante la función header, y redirigimos al usuario hacia la página principal. (hubiese sido posible redirigir también hacia una página de error 404 para informar al usuario).

4.2.2 Lang Engine

Este es el modulo encargado de cargar el idioma de la página, se apoya en las variables de sesión obtenidas en el Url Engine y que nos indican cual es el idioma de la url. También se encarga de generar el menú para cambiar de idioma la pagina y el código "hreflang" este es un código que se añade al <head> y que se utiliza sobre todo para mejorar SEO, ya que con el indicamos a google el idioma en que nos encontramos y cuáles son sus respectivas urls en otros idiomas. (evita que nos detecte contenido duplicado)

```

<?PHP
namespace langEngine;

/*LANG ENGINE*/

if(!$_SESSION['ui_lang']){
    $_SESSION['ui_lang']="es";
}

$thisPageLang=$_SESSION['ui_lang'];

$_SESSION['ui_currency']=1;

global $lang;
include_once('./modules/lang_engine/languages/' . $_SESSION['ui_lang'] . '.php');

$sw->addVars($lang, "uiLang");
log_msg($sw->uiLang, 'Diccionario!');

```

Su funcionamiento es simple, primero verificamos que esta creada la variable de sesión "ui_lang" si no esta creada la creamos con el idioma que queremos asignarle por defecto, en este caso "es".

También creamos una variable de sesión "ui_currency" , le damos valor uno que es el valor de la moneda en nuestra base de datos, en este caso el euro, aunque aún no está implementado la idea es que a la larga la web sea capaz de mostrarse también en varias monedas, por ello hay realizadas ya algunas opciones pensando en el futuro.

Después creamos una variable global que será \$lang e incluimos el archivo del diccionario del idioma correspondiente.

Este archivo hace uso de la variable global lang y crea un array con todas las palabras o frases que nos van haciendo falta. Podemos incluir cuantas queramos siempre y cuando incluyamos también sus traducciones en los demás idiomas.

```

$lang['yes']="si";
$lang['no']="no";
$lang['long_title']="Cursos de larga duraci@acute;n";
$lang['included']="incluido";
$lang['not_included']="no incluido";
$lang['lang_select']="idioma";
$lang['cur_select']="moneda";
$lang['view_map']="ver mapa";
$lang['booking_tool_title']="precio n@acute;nno garantizado";
$lang['booking_tool_summary']="resumen de la reserva";
$lang['booking_now']="reserva tu curso ahora";
$lang['includes']="incluye";
$lang['booking_tool_lessons_x_w']="lecciones por semana";
$lang['booking_tool_more_info']="+ M@acute;s info $xacute;n";
$lang['booking_tool_less_info']="Menos informaci@acute;n $xacute;n";
$lang['booking_tool_minimum_age']="edad n@acute;n";
$lang['booking_tool_course_hours']="Test de n@acute;n";
$lang['booking_tool_course_days']="d@acute;as del curso";
$lang['booking_tool_max_stu_x_class']="n@acute;n de alumnos";
$lang['booking_tool_school_admin_fee']="Cuota admin. escuela";
$lang['booking_tool_average_num_stu']="***";
$lang['booking_tool_lesson_duration']="duraci@acute;n";
$lang['booking_tool_course_material']="material did@acute;tico";
$lang['booking_tool_course_cert']="Certificado final";
$lang['booking_tool_course_level']="niveles";
$lang['booking_tool_starting_date']="comienza";
$lang['booking_tool_course_weeks']="semanas";
$lang['booking_tool_week']="semana";
$lang['booking_tool_empty_reservation']="Selecciona uno de los cursos";
$lang['booking_tool_stu_x_room']="personas por habitaci@acute;n";
$lang['booking_tool_kitchen']="cocina";

```

Una vez lo tenemos cargado añadimos la variable `$lang` a nuestro objeto, para que nos sea más sencillo y rápido trabajar con ellas.

Podemos observar que muchas de las cosas que vamos añadiendo las incluimos siempre en el objeto `$sw`, esto nos permitirá además de trabajar mas cómodamente con toda la información, una mejor respuesta en cuanto a velocidad de procesamiento, porque tendremos toda la información cargada ya en el objeto.

Por último construimos el menú de cambio de idiomas , buscando la url actual , mostrando el idioma como idioma activo y buscando las urls en otros idiomas, su construcción es sencilla mediante la creación de una lista. Lo mismo hacemos con el código de `hreflang`.

Una vez contruidos los añadimos también al objeto `$sw` para luego ser llamado y aplicado en el lugar que queramos.

4.2.3 Lang Menu

Este modulo aunque se carga en todas las urls ahora mismo solo se utiliza en la homepage, pero en su momento en la realización pensamos que podía ser mejor cargarlo siempre, porque igual nos interesaba mostrarlo más adelante en otras de las urls.

Básicamente lo que hace este modulo es construir un menú con los idiomas que tenemos disponibles para estudiar y los destinos, divididos en países y ciudades. El menú como habíamos decidido en la maquetación se mostrará en ventanas modales. Es decir ventanas emergentes que aparecerán cuando seleccionemos alguna de las opciones.

Su construcción se basa en recorrer los idiomas de los que disponemos , los países en los que se imparten y las ciudades de esos países en las que tenemos escuelas.

A partir de aquí y con la ayuda de varios `foreach()` construimos el código html de las ventanas modales.

```

foreach($languages as $k=>$v)
{
    $lang_id=$v['lang_id'];
    $sta=$sw->goFast('languages','id',$lang_id,'status');
    if($sta=="1"){
        $lan=$sw->goFast('languages','id',$lang_id,'lang');
        $name=$v['name_'.$_SESSION['ui_lang']];

        $lang_html.='<div class="flag" id="flag_'.$x.'">
<a href="#" data-toggle="modal" data-target="#'.$lan.'" id="">

<div class="flag-text">'.ucfirst($name).'</div>
</a>
</div>';

        $modal_country_html.='<div class="modal fade" id="'.$lan.'">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">
<button type="button" class="close" data-dismiss="modal" aria-hidden="true">&times;</button>
<h3 class="modal-title">'.ucfirst($sw->uiLang['search_destination']).'</h3>
</div>
<div class="modal-body">

```

Podemos observar que en este punto ya comenzamos a hacer uso del diccionario del idioma cargado en el objeto \$sw.

```

$modal_country_html.='<div class="modal fade" id="'.$lan.'">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">
<button type="button" class="close" data-dismiss="modal" aria-hidden="true">&times;</button>
<h3 class="modal-title">'.ucfirst($sw->uiLang['search_destination']).'</h3>
</div>
<div class="modal-body">

```

La función "ucfirst" que la veremos también en muchas ocasiones a lo largo de todo el código, es una función de PHP que nos permite poner la primera letra de un string en mayúsculas. También es posible que veamos "ucwords" que nos pone la primera letra de cada palabra de un string en mayúscula. Estas funciones son de gran ayuda para controlar todo el contenido escrito y que siga una concordancia y este redactado de manera correcta.

4.2.4 Meta tags module

En un principio este modulo era solo pensado para construir los el código de meta tags para seo necesario en cada url, y esta es su principal función, pero se ha añadido otras, ya que al ser un modulo que queríamos utilizar luego dentro de la etiqueta <head> se ha pensado que podía ser aprovechado para incluir otros códigos que también son necesarios, como por ejemplo el código de seguimiento de Analytics, incluir el código del chat online, y también para construir el código de "breadcrumb" o "miga de pan" que luego incluiremos y mejorará la experiencia de navegación de usuario.

Comenzamos a construir el código html dentro de la variable \$metashtml, entre los que incluimos metas de lenguaje, distribución, código de analytics, title, description y metas específicas para los robots de búsqueda.

```

$metashtml=<meta http-equiv="Content-Language" content="'.$_SESSION['ui_lang'].'"/>
<meta name="distribution" content="global"/>;

$metashtml.="<script type=\"text/javascript\">

var _gaq = _gaq || [];
_gaq.push(['_setAccount', 'UA-1691810-1']);
_gaq.push(['_setCampNameKey', 'utm_name']);
_gaq.push(['_trackPageview']);
setTimeout(function() {
  var ga = document.createElement('script'); ga.type = 'text/javascript'; ga.async = true;
  ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') + '.google-analytics.com/ga.js';
  var s = document.getElementsByTagName('script')[0]; s.parentNode.insertBefore(ga, s);
})();
</script>;

```

Todas las metas son obtenidas mediante una consulta a la base de datos donde contamos con el title, y el description de cada url por separado. Esto nos permite un control y un mejor mantenimiento de cada uno de los metas para cada url.

4.2.5 Footer

EL modulo footer como su propio nombre indica es el encargado de construir el footer de la web. Básicamente lo que realiza es consultas a la base de datos para obtener las urls de las secciones que queremos que aparezcan en el footer y en el idioma correspondiente.

Y después construimos el html en la variable \$footer.

```

$quienes_url=$sw->readRow("site_urls",array("type"=>"about","lang_id"=>$sw->goFast("languages","lang",$SESSION['ui_lang']));
$contact_url=$sw->readRow("site_urls",array("type"=>"contact","lang_id"=>$sw->goFast("languages","lang",$SESSION['ui_lang']));
$terminos_url=$sw->readRow("site_urls",array("type"=>"terms","lang_id"=>$sw->goFast("languages","lang",$SESSION['ui_lang']));
$price_url=$sw->readRow("site_urls",array("type"=>"price","lang_id"=>$sw->goFast("languages","lang",$SESSION['ui_lang']));

$footer='
<div class="container" style="margin-bottom:0px;">

<div class="span10 offset1 text-center">

<footer style="margin: 0; margin-top:40px;">

<ul class="nav nav-justified text-center">

<li><a style="height: 60px;" href="'. $site.'.$quienes_url['full_uri'].'">'.ucfirst($sw->uiLang['about_title']).'</a></li>
<li><a style="height: 60px;" href="'. $site.'.$contact_url['full_uri'].'" tabindex="100">'.ucfirst($sw->uiLang['menu_contact']).'</a></li>
<li><a style="height: 60px;" href="'. $site.'.$terminos_url['full_uri'].'" tabindex="100">'.ucfirst($sw->uiLang['footer_terms']).'</a></li>
<li><a style="height: 60px;" href="'. $site.'.$price_url['full_uri'].'" tabindex="100">'.ucfirst($sw->uiLang['booking_tool_title']).'</a></li>

<li><a style="height: 60px;" href="http://blog.quality-courses.com" >Blog</a></li>
<li><!-- AddThis Follow BEGIN -->
<div style="width: 180px;" class="addthis_toolbox addthis_32x32_style addthis_default_style">
<a class="addthis_button_facebook_follow" addthis:userid="QualityCourses.LanguagesAbroad?ref=ts"></a>
<a class="addthis_button_twitter_follow" addthis:userid="QualityCourses"></a>
<a class="addthis_button_linkedin_follow" addthis:userid="445831?xk=vwah&trkInfo=tag%3Aqualitycourses%2Cid%3A1-1-1" addthis:usertype="company"></a>
<a class="addthis_button_google_follow" addthis:userid="/0/+qualitycourses/posts"></a>
<a class="addthis_button_youtube_follow" addthis:userid="QualityCoursesVideo"></a>

```

Por último utilizamos la sentencia "\$xtpl->assign('footer',\$footer);" que también veremos en mas casos a partir de ahora y lo que hace es que asigna una variable

al objeto de la herramienta xTemplate para luego ser llamada desde el html con el nombre que le indicamos.

Sera explicado con más detalle más adelante para entender el funcionamiento de esta herramienta.

4.2.6 Booking tool

La herramienta quizás mas compleja, ya que es la encargada de la función del carrito de compra, recoge datos de la base de datos de escuelas, precios, cursos y alojamientos y además trabaja con fechas y con los precio. En definitiva montar toda la herramienta de reserva y construye el html de cursos y alojamientos para cada una de las escuelas.

Este modulo va acompañado de un archivo functions.php creado para separar algunas de las funciones y tenerlo más ordenado, y es lo primero que incluimos para tenerlo a nuestra disposición para cuando necesitemos algunas de las

```
require_once(MAIN_PATH.MODULES_PATH.'/booking_tool/functions.php');

switch($_SESSION['ui_lang'])
{
    case "en":$date_format="d/m/Y";break;
    default:$date_format="d/m/Y";
}

//fechas
if(!isset($_GET['duration'])){ $_GET['course_duration']="2"; $fun=""; }else{$_GET['course_duration']=$_GET['duration'];
$fun="carruaje()";
carraje():";
}
$xtpl->assign('fun',$fun);
if(!isset($_GET['course_std'])){ $_GET['course_std']=date($date_format,strtotime('next week next monday')); }
$xtpl->assign('course_std',$_GET['course_std']);
if($_SESSION['ui_lang']!="en"){ $_GET['course_std']=format_date($_GET['course_std'],"/","es"); }

$c_std=format_date($_GET['course_std']);

//inicio alojamiento
$_GET['acc_std']=date($date_format,strtotime("-1 day",strtotime($c_std)));

log_msg($_GET['course_std'],'INICIO CURSO');
log_msg($_GET['acc_std'],'INICIO ALOJAMIENTO');

$acc_before=date($date_format,strtotime("-2 day",strtotime($c_std)));
```

funciones que tengamos definidas en el.

Lo primero que hace es controlar lo que nos llegar por GET, para poder ofrecer al usuario la información correcta que viene buscando.

Por GET pueden llegarnos varias cosas, como la fecha de inicio del curso "course_std" la duración en semanas "course_duration" ...etc

También establecemos unos valores por defecto en el caso de que no nos lleguen variables por GET, por ejemplo la duración por defecto es la mínima para la que vendemos cursos que son 2 semanas y el día de comienzo es el siguiente lunes al

día actual por defecto. Esto lo conseguimos con la ayuda de funciones de PHP como *date* o *strtotime*.

Para asegurarnos un buen formato de fecha utilizamos una de las funciones que tenemos en nuestro *functions.php*, que sería *format_date* y que se encarga de ponernos la fecha en formato *día/mes/año*.

La fecha de inicio de alojamiento siempre la calculamos automáticamente, un día antes de la fecha de inicio del curso.

```
//fin del curso

$dur=$_GET['course_duration'];

log_msg($dur,'DURATIONN');

$_GET['course_end']=date($date_format,strtotime("+".$_GET['course_duration']." week",strtotime($c_std)));
log_msg($_GET['acc_ed'],'FIN ALOJAMIENTO');
log_msg($_GET['course_end'],'FIN CURSO');

//dia antes
$acc_after=date($date_format,strtotime("+".$_GET['course_duration']." week",strtotime("-2 day",strtotime($a_std))));

//fin fechas

$school_data=$sw->readRow("schools_data","id:".$sw->pageVars['ref']);
```

Las fechas de finalización también son calculadas automáticamente en base a la duración que se haya escogido.

Una vez establecida todas las fechas pasamos a la lectura de la información de la escuela en concreto, con estos datos luego trabajaremos tanto para mostrar la información, fotos...etc de la escuela como para acceder también a las consultas de los cursos, alojamientos y precios.

```
if($school_data)
{
    $rating=$school_data['rating'];
    $rating=($rating*75)/5;
    $xtpl->assign('rating',$rating);
    $xtpl->assign('ratingss',$school_data['rating']);

    $xtpl->assign('info',ucfirst($lang['school_general_information_t']));
    $school_name=$school_data['school_name'];
    $school_description=$school_data['school_description_'.$_SESSION['ui_lang']];
    $xtpl->assign('school_name',ucfirst($school_name));
    $xtpl->assign('school_description',ucfirst(stripslashes($school_description)));
    $school_address=$school_data['school_address'];
    $xtpl->assign('school_address',ucfirst($school_address));
    $xtpl->assign('ui_lang_id',$_SESSION['ui_lang_id']);
    $xtpl->assign('ui_lang',$_SESSION['ui_lang']);
}
```

Trabajamos con los datos obtenidos de la escuela y los asignamos también al objeto de xTemplate como vemos en la captura del código, obtenemos toda la información de la escuela en concreto.

Se construye también el html las instalaciones de con las que cuenta la escuela y que también las obtenemos de los datos de la escuela en base de datos.

```

if($school_data['wifi']==1){
    $instalaciones.='<li>
        <div class=" " style="float: left; color: #000000;">
        <button id="button" type="button" style="padding:0px; height: 40px; width: 40px; text-align: center;" class="btn" data-toggle="popover"
        
        </button>
        </div>
        </li>';
        $control=1;
    }
if($school_data['coffe_bar']==1){
    $instalaciones.='<li>
        <div class="" style="float: left; color: #000000;">
        <button id="button" type="button" style="padding:0px; height: 40px; width: 40px; text-align: center;" class="btn" data-toggle="popover"
        
        </button>
        </div>
        </li>';
        $control=1;
    }
if($school_data['library']==1){
    $instalaciones.='<li>
        <div class="" style="float: left; color: #000000;">
        <button id="button" type="button" style="padding:0px; height: 40px; width: 40px; text-align: center;" class="btn" data-toggle="popover"
        
        </button>
        </div>
        </li>';
    }

```

Otra de las funciones interesantes de comentar en este archivo es con la cual cambiamos los enlaces de los videos de cada escuela de youtube. En la información de la escuela en base de datos tenemos almacenado la url del enlace de los videos en youtube. Y mediante esta función conseguimos extraer la id del video y montar una nueva url de youtube para asegurarnos de que cargan perfectamente.

```

function youtubeId($url) {
if($url != '') {
    $match = preg_match("#(?<v=) [a-zA-Z0-9-]+ (?=&) | (?<v\/) [^\n]+ (?=\?) | (?<v=) [^\n]+ | (?<youtu.be\/) [^\n]+#", $url, $matches);
    if((int) $match > 0) {
        return $matches[0];
    }
}
return false;
}

$videoid=youtubeId($school_data['video']);
log_msg($videoid,'VIDEOIDDD');
log_msg($school_data['video'],'VIDEO');
if($videoid!=FALSE){
    $video='<a href="http://www.youtube.com/watch?v='.$videoid.'" data-height="480" class="lightbox btn btn-info" ><span class="glyphicon">
    }
else{$video='';}

$xtpl->assign('video',$video);

```

Lectura de carpeta de imágenes.

```
$moreimages='';  
/*Las demas imagenes*/  
$handle=opendir('./images/escuelas/'.$school_data['images_folder']);  
$name=array();  
while($name=readdir($handle))  
{  
    if($name!="." && $name!=".."){ $school_images[]=$name;  
    $moreimages.='<a href="'.$sw->coreOptions['site_url'].'/images/escuelas/'.$school_data['images_folder'].'/.'.$name.'" data-group="set" title="'  
    }  
}  
closedir($handle);  
  
log_msg($school_images,'Images');  
$tpl->assign('moreimages',$moreimages);
```

Estas líneas de código nos permiten abrir un directorio y recorrer su contenido, con el construimos el código html que mostrará las imágenes de cada escuela. Lo guardamos en la variable *\$moreimages* y luego la cargamos en el objeto de xTemplate para poder llamarla desde html.

En este archivo construimos también el código del mapa de localización, haciendo uso de la API V3 de Google Maps, incluimos el script y generamos los marcadores de escuela obteniendo de la base de datos los datos necesarios como latitud, longitud.

```
/*ADD MARCADORES*/  
$mapa.="  
    //MARCADOR  
    var myLatLng = new google.maps.LatLng("$.school_data['add_lat'].","$.school_data['add_lon'].");  
  
    var infowindow".$numb." = new google.maps.InfoWindow({  
    content: "$.school_data['school_name']."});  
  
    var marker".$numb." = new google.maps.Marker({  
    position: myLatLng,  
    map: map,  
    title: "$.school_data['school_name']."}  
});  
  
    google.maps.event.addListener(marker".$numb.", 'click', function() {  
    infowindow".$numb.".open(map,marker".$numb.");  
});  
";  
$mapa.="";  
  
$mapa.="google.maps.event.addDomListener(window, 'load', initialize);";  
  
$mapa.="</script>";
```

Otro de los puntos a tener en cuenta dentro del código de este archivo es la recopilación de datos sobre cursos, alojamientos y trasportes de la escuela en concreto. Con ellos pasaremos a construir la parte de venta de la pagina de

escuela, donde el usuario puede ver todos los cursos disponibles, tipos de alojamientos, precios y añadirlos a su carro de compra.

```
//DATOS DE LOS CURSOS DE LA ESCUELA
$order="ORDER BY 'lessons_x_week' ASC";
$course_data=$sw->readTable("courses_data",array("school_id"=>$sw->pageVars['ref'], "status"=>1), $order);

log_msg($course_data, 'COURSE DATA');
$SESSION['school_id']=$course_data[0]['school_id'];
log_msg($SESSION['school_id'], 'SCHOOL ID');
//DATOS DE LOS ALOJAMIENTOS
$sacc_data=$sw->readTable("accommodations_data",array("school_id"=>$sw->pageVars['ref'], "status"=>1));
log_msg($sacc_data, 'ACC DATA');
//DATOS DE TRANSPORTES AEROPUERTO
$air_pickup_data=$sw->readTable("airport_pickup",array("school_id"=>$sw->pageVars['ref'], "status"=>1));
log_msg($air_pickup_data, 'AIR PICKUP DATA');
```

Antes de la construcción del html con cursos y alojamientos, una cosa más a destacar es calculo de suplementos, en el cual mediante dos fechas de inicio y fin , podemos ver cuando comienza y acaba la alta temporada , ya que algunas escuelas aplican un suplemento económico a sus precios por semana.

Por ello es necesario calcular mediante las fechas de inicio y fin del curso cuantas semanas entran dentro del periodo de suplemento y por tanto cuantos suplementos se han de sumar al precio. Las fechas de inicio y fin de alta temporada las obtenemos de la información de la escuela, y los precios y el valor del suplemento lo obtenemos de la tabla de precios.

Esta tarea se repite también con los alojamientos que también tengan esta diferencia de precios en temporada alta.

```
/*CALCULO DE SUPLEMENTOS*/
$mas=0;

$sup_std=$_GET['course_std'];
$scsup_std=format_date($sup_std);
if (strtotime($scsup_std) >= strtotime($f_inicio) and strtotime($scsup_std) <= strtotime($f_fin)){

    $mas=$mas+1;
    $dur=$dur-1;

}
While($dur>0){

$sup_std=date($date_format,strtotime("+1 week",strtotime($scsup_std)));
$scsup_std=format_date($sup_std, "/", "en");

log_msg($scsup_std, 'Inicio SEMANA');
log_msg($f_inicio, 'Inicio SUPLEMENTO');

if (strtotime($scsup_std) >= strtotime($f_inicio) and strtotime($scsup_std) <= strtotime($f_fin)){
$dentro="dentro";
log_msg($dentro, 'SEMANA SUPLEMENTO');
$mas=$mas+1;
}
```


Para ello mediante ajax, al que dedicaremos una sección exclusiva en esta memoria, utilizamos el archivo carro.php para formar el carro y también crear el resumen del pedido, recargando la información sin cambiar de pagina.

4.3 Home

En este punto se comienza a trabajar en el código de la pagina home, administrado por el archivo home-tpl.php. Lo primero que se hace en este archivo es cargar el template que queremos para la home, y luego incluimos los modules que necesitamos y no tengamos incluidos ya.

```
1 <?PHP
2
3 /*TEMPLATE*/
4 $xtpl = new XTemplate('./content/templates/home.xtpl'); //Plantilla
5 log_msg('Template Cargado...');
6
7 //Modulo Meta Tags
8 include('./modules/meta_tags.php');
9
```

Entre algunos de los modules que se cargan están el "meta_tags" o el "footer", también como en la pagina se mostraran algunas escuelas con precios y datos, cargamos las funciones que tenemos dentro del modulo "booking tool" para poder hacer uso de ellas en caso de que sean necesarias.

En esta página se tiene que cargar también el menú de idiomas que se pueden estudiar y que cargara mediante ventanas modales los países y las ciudades donde se tienen escuelas que se pueden comprar a través de la web.

```
12 /*config*/
13 $xtpl->assign('SITE_URL',$sw->coreOptions['site_url']);
14 $site=$sw->coreOptions['site_url'];
15 /*MENU*/
16 $xtpl->assign('menu_contact',ucfirst($sw->uiLang['menu_contact']));
17 /*lang engine*/
18 $xtpl->assign('lang_engine',$sw->lang_module);
19 $xtpl->assign('hreflang',$sw->hreflang);
20 $xtpl->assign('this_lang',ucfirst($sw->this_lang));
21 /*lang_menu*/
22 $xtpl->assign('lang_menu',$sw->lang_menu);
23 $xtpl->assign('modales_country',$sw->modales_country);
24 $xtpl->assign('modales_city',$sw->modales_city);
```

Y a continuación comenzamos a cargar frases o palabras que tenemos en nuestro diccionario de idioma y que utilizaremos en esta página. Cada una de las variables que queremos llamar luego desde el html y queremos que aparezcan son asignadas con un nombre al objeto \$xtpl que es el objeto del xTemplate.

```

26 //Modulo Footer
27 include('./modules/footer.php');
28
29 require_once(MAIN_PATH.MODULES_PATH.'/booking_tool/functions.php');
30
31 /*Comenzamos*/
32 $xtpl->assign('title',ucfirst($sw->uiLang['sub_title']));
33 $xtpl->assign('sub_title',ucfirst($sw->uiLang['sub_title2']));
34 $xtpl->assign('destacadas',ucfirst($sw->uiLang['destacadas']));
35 $xtpl->assign('from',$sw->uiLang['from']);
36 $xtpl->assign('menu_contact',ucfirst($sw->uiLang['menu_contact']));
37 $xtpl->assign('menu_idioma',$sw->uiLang['menu_idioma']);
38 $xtpl->assign('home1',strtoupper($sw->uiLang['home1']));
39 $xtpl->assign('home2',strtoupper($sw->uiLang['home2']));
40 $xtpl->assign('home3',strtoupper($sw->uiLang['home3']));
41 $xtpl->assign('lessons_x_w',ucfirst($sw->uiLang['booking_tool_lessons_x_w']));
42 $xtpl->assign('weeks',$sw->uiLang['booking_tool_course_weeks']);
43 $xtpl->assign('more',$sw->uiLang['booking_tool_more_info']);
44 $xtpl->assign('more_details',ucfirst($sw->uiLang['more_details']));
45 $xtpl->assign('calidad_precio',ucfirst($sw->uiLang['calidad_precio']));
46 $xtpl->assign('reservar',ucfirst($sw->uiLang['booking_tool_book_now_r']));
47 $xtpl->assign('numerotlf',$sw->uiLang['numerotlf']);
48
49
50
51 $xtpl->assign('long_title',ucfirst($sw->uiLang['long_title']));
52 $xtpl->assign('workandstudy_title',ucfirst($sw->uiLang['workandstudy_title']));
53 $xtpl->assign('menu_busines',ucfirst($sw->uiLang['menu_busines']));
54 $xtpl->assign('menu_juniors',ucfirst($sw->uiLang['menu_juniors']));
55

```

Y también cargamos las urls de las secciones que aparecen en la home.

```

60 $long_url=$sw->readRow("site_urls",array("type"=>"long","lang_id"=>$sw->goFast("languages","lang",$_SESSION['ui_lang']));
61
62 $xtpl->assign('long_url',$site.$long_url['full_uri']);
63
64 $work_url=$sw->readRow("site_urls",array("type"=>"work","lang_id"=>$sw->goFast("languages","lang",$_SESSION['ui_lang']));
65
66 $xtpl->assign('work_url',$site.$work_url['full_uri']);
67
68 $business_url=$sw->readRow("site_urls",array("type"=>"business","lang_id"=>$sw->goFast("languages","lang",$_SESSION['ui_lang']));
69
70 $xtpl->assign('business_url',$site.$business_url['full_uri']);
71
72
73 if($_SESSION['ui_lang']!="es"){ $url_camp="http://www.internationalsummercamps.com"; $xtpl->assign('camp_url',$url_camp); } else{
74 $url_camp="http://www.campamentos-de-verano.com"; $xtpl->assign('camp_url',$url_camp);
75 }
76

```

Por último en este archivo cabe destacar la selección de las escuelas destacadas que queremos que aparezcan en la Home, dependiendo del idioma de la pagina que utilicemos cargamos unas escuelas destacadas distintas.

Recorremos el array de escuelas destacadas consultando en la base de datos la información de cada una de las escuelas y sacando de cada una el precio mínimo que es el de dos semanas y es el que queremos mostrar. Cada escuela la mostramos gráficamente en una caja y para ello se utiliza la función *parse* de *xTemplate* que permite hacer una iteración a la hora de pintar código html.

```

94 /*DESTACADAS HOME*/
95 /*4 ciudades destacadas en la home*/
96 if($_SESSION['ui_lang']=="es"){
97 $escuelas_dest = array("96","111","36","90");
98 }
99 if($_SESSION['ui_lang']=="en"){
100 $escuelas_dest = array("96","111","36","90");
101 }
102 if($_SESSION['ui_lang']=="fr"){
103 $escuelas_dest = array("121","96","111","23");
104 }
105 if($_SESSION['ui_lang']=="it"){
106 $escuelas_dest = array("121","96","111","36");
107 }
108 if($_SESSION['ui_lang']=="de"){
109 $escuelas_dest = array("121","96","111","36");
110 }
111 if($_SESSION['ui_lang']=="es"){
112 $escuelas_dest = array("121","96","111","36");
113 }
114

```



```

215     $xtpl->parse('main.hometable.homerow.homeItem');
216
217
218
219     }
220     $xtpl->parse('main.hometable.homerow');
221     $xtpl->parse('main.hometable');

```

Aquí el parte del código html que forman el cuadro de escuela, mediante una sentencia iterativa como *while* o *for* y la función *parse* se pueden pintar cuantos cuadros se quieran.

```

<div class="row" style="margin-top: 25px;">
  <!-- BEGIN: hometable -->
  <!-- BEGIN: homerow -->
  <!-- BEGIN: homeItem -->
  <div class="col-md-3 col-sm-6 col-xs-12">
    <a href="{school_url}"><div class="thumbnail gaja" style="height: 410px;">
      <a href="{school_url}">
        
      </a>
      <span class="label label-success"><div class="content" style="font-size: 14px; font-weight: 300;"><div class="col-md-6"
    <div class="caption">
      <h3 class="title" style="font-size: 17px; margin-top:15px; margin-bottom:0px; color:#f1592a; font-weight: 400;"><a href="{city_url}">{nombre_ciudad}</a></h3>
      <h4 style="margin-top:5px; font-size: 14px;"><a href="{city_url}">{nombre_ciudad}</a></h4>
      <div class="ratings">
        <!--<p class="pull-right">6 reviews</p-->
      <span class="ui-rater">
    <span class="ui-rater-starsOff" style="width: 75px; cursor: pointer; opacity: 1;margin-top: 10px; margin-bottom: 10px;"><span class="ui-rater-starsOn" style="width: 75px; cursor: pointer; opacity: 1;margin-top: 10px; margin-bottom: 10px;">
  </span>
</div>
  <ul class="list-unstyled">
    <li><span><strong>{calidad_precio}</strong></span></li>
  </ul>

```

Se puede ver como en el código html se indica donde comienza la tabla, la fila o el elemento que queremos repetir y utilizar con las función *parse*, con la línea *<!-- BEGIN: hometable -->*

Esta forma de escribir el html de manera dinámica a medida que lo necesitamos es una de las ventajas de utilizar xTemplate para la carga del HTML y el PHP por separado y será utilizada en otras secciones de la web.

4.4 Ciudad / País

Para el funcionamiento de las páginas de ciudad y país aunque se utilizan dos archivos php por separado (country-tpl.php y city-tpl.php) su funcionamiento es similar, como en el caso anterior lo primero que hacemos es cargar la plantilla, los módulos y textos que necesitaremos. Otra de las ventajas de trabajar con xTemplate y las plantillas es que como en este caso, ambas paginas sea de país o ciudad utilizan la misma plantilla (resultados.xtpl).

```

3  /*TEMPLATE*/
4  $xtpl = new XTemplate('./content/templates/resultados.xtpl'); //Plantilla
5  log_msg('Template Cargado...');
6
7  /*config*/
8  $xtpl->assign('SITE_URL', $sw->coreOptions['site_url']);
9  $site=$sw->coreOptions['site_url'];
10 /*MENU*/
11 $xtpl->assign('menu_contact', ucfirst($sw->uiLang['menu_contact']));
12 /*lang engine*/
13 $xtpl->assign('lang_engine', $sw->lang_module);
14 $xtpl->assign('hreflang', $sw->hreflang);
15 $xtpl->assign('this_lang', ucfirst($sw->this_lang));
16 /*lang menu*/
17 $xtpl->assign('lang_menu', $sw->lang_menu);
18
19 /*contact URL*/
20
21 $contact_url=$sw->readRow("site_urls", array("type"=>"contact", "lang_id"=>$sw->goFast("languages", "lang", $_SESSION['ui_lang'])););
22 log_msg($contact_url, "HOME URL");
23 $xtpl->assign('contact_url', $site.$contact_url['full_uri']);
24 //Modulo Footer
25 include('./modules/footer.php');

```

En este caso como en el anterior también se hace uso del archivo functions.php que está dentro del modulo *booking tools* para utilizar algunas de sus funciones por ejemplo en la elección de fechas y a la hora de trabajar con precios.

La principal función de esta página es la de mediante la *ref* que tenemos en la *url* nos indica la id de la ciudad o país, extraemos de la base de datos todos los datos del destino al que corresponda, después extraemos también todas las escuelas con las que se cuenta en dicho destino, mostrando, duración del curso, precio, puntuación de la escuela, foto...etc.

Estas páginas funcionan como un buscador, al cual le aplicamos filtros y mediante ellos se van cambiando los resultados obtenidos en función de la elección del usuario. Todo este proceso se realiza mediante llamadas *ajax* con *jquery* que como se ha dicho con anterioridad será explicado más detalladamente en un apartado de esta memoria exclusivo para ello.

```

112 //EMPIEZA CITY
113 $city_data=$sw->readRow("cities_data", "id:".$sw->pageVars['ref']);
114
115 if($city_data)
116 {
117
118
119     /*Others*/
120
121     $id_country=$sw->goFast("city_names", "id", $city_data['city_id'], "id_country");
122     $others=$sw->readTable("city_names", array("id_country"=>$id_country, "status"=>1));
123     log_msg($others, 'OTHERS CITIES');
124
125     $other_html='';
126     foreach($others as $o=>$nn) {
127
128         log_msg($nn, 'NN');
129         $other_city_name=$sw->goFast("cities_translate", "id_city", $nn['id'], "value_".$_SESSION['ui_lang']);
130         if (!$other_city_name) {$other_city_name=$sw->goFast("city_names", "id", $nn['id'], "value");}
131         $id_city=$sw->goFast("cities_data", "city_id", $nn['id'], "id");
132         $other_city_url=$sw->readRow("site_urls", array("type"=>"city", "ref"=>$id_city, "lang_id"=>$sw->goFast("languages", "lang", $_SESSION['ui_lang'])););
133         $url_other=$sw->coreOptions['site_url'].$other_city_url['full_uri'];
134         // $sw->assign('school_url', $sw->coreOptions['site_url'].$school_url['full_uri']);
135         $other_html.=<a href="$url_other" class="btn btn-primary" onClick="$sw->pushEvent('click menu ciudad o país', 'click menu ciudad o país', 'click menu ciudad o país')">
136

```

Como podemos observar en la captura, lo primero que se hace como hemos dicho es extraer los datos de la ciudad `$city_data` con una consulta a la base de datos buscando la id que coincida con la ref de la url de la ciudad.

Inmediatamente después lo que hacemos es crear un menú con otros destinos dentro del mismo país donde se estudia el idioma que en el que se está interesado.

Otro punto interesante es la creación dinámicamente de lo que será el título `<h1>` dentro de la página, que se genera automáticamente con una frase del tipo "Cursos de inglés en Londres" sustituyendo el idioma y el destino por el correspondiente en cada caso.

```
204 if($_SESSION['ui_lang']=='en'){ $content_title=ucwords($study_language[0]). " %courses% %in% ".ucwords($city_name). " ";
205 $title_search=ucfirst($sw->uiLang['search_title']). " %in% ".ucwords($city_name);}
206
207 if($_SESSION['ui_lang']=='es'){ $content_title="%courses% ".ucwords($study_language[0]). " %in% ".ucwords($city_name). " ";
208 $title_search=ucfirst($sw->uiLang['search_title']). " %in% ".ucwords($city_name);}
209
210 if($_SESSION['ui_lang']=='de'){ $content_title="%courses% %in% ".ucwords($city_name). " ";
211 $title_search=ucfirst($sw->uiLang['search_title']). " %in% ".ucwords($city_name);}
212
213 if($_SESSION['ui_lang']=='fr'){ $content_title="%courses%.ucwords($study_language[0]). " %in% ".ucwords($city_name). " ";
214 $title_search=ucfirst($sw->uiLang['search_title']). " %in% ".ucwords($city_name);}
215
216 if($_SESSION['ui_lang']=='it'){ $content_title="%courses% ".ucwords($study_language[0]). " %in% ".ucwords($city_name). " ";
217 $title_search=ucfirst($sw->uiLang['search_title']). " %in% ".ucwords($city_name);}
218
```

Para ello, dependiendo del idioma en que nos encontremos montamos un *string* de texto con algunas partes del texto entre signos de "%" que luego serán sustituidos mediante la función de *php "str_replace"* que nos permite reemplazar el texto que le indiquemos dentro de un *string*. Hacemos lo mismo con otros títulos de la página como el `<h3>`.

```
$h3_title_final=str_replace("%country%",ucwords($city_name),$h3_title_final);
```

En estas páginas también se utiliza la *API de Google maps V3* para generar el mapa con todas las escuelas disponibles en esa ciudad o país en concreto.

Se hace una consulta a la base de datos para sacar el total de escuelas del destino y todos sus datos. Entre ellos los datos de latitud y longitud y se genera el código javascript del mapa dinámicamente.

```
$consulta="SELECT DISTINCT `schools_data`.`id`, `schools_data`.`video`, `schools_data`.`school_description_es`, `schools_data`.`school_address`,
`schoools_data`.`images_folder`, `schools_data`.`main_image`, `schools_data`.`school_name`,`schools_data`.`comision`,`courses_data`.`course_type`,
`courses_data`.`lessons_x_week`, `course_prices`.`$. $dura."w` as c price, `course_prices`.`currency_id`,`course_prices`.`admin_fee`, `cities_data`.`
JOIN `courses_data` ON `schools_data`.`id` = `courses_data`.`school_id` JOIN `course_prices` ON `courses_data`.`id` = `course_prices`.`course_id`
JOIN `cities_data` ON `schools_data`.`city_id` = `cities_data`.`city_id`
WHERE `cities_data`.`id` = '". $sw->pageVars['ref']."' AND `schools_data`.`status` = '1' AND `courses_data`.`status` = '1'
AND `course_prices`.`$. $dura."w` > 0.00 ORDER BY `course_prices`.`2w` ASC";
```

En esta consulta mySql puesto que trataremos con datos de varias tablas que están relacionadas entre sí utilizamos la sentencia JOIN para establecer estas relaciones y optimizar al máximo el tiempo de las consultas. Como vemos esta consulta depende también de la *ref* obtenida en la *url* que nos indica el destino y la duración para la que queremos mostrar precios de las escuelas (por defecto 2 semanas).

Con esta consulta contamos el total de resultados para obtener el total de escuelas que tendremos y luego poder dividir la consulta para que no se muestren desde un principio todas las escuelas.

```
$data=$sw->sqlToArray($consulta); # return a 2 dimensions array
$_SESSION['rows']=$sw->countRows();
$resulcon=$_SESSION['rows'];
```

Para ello utilizamos la función que de *PtcDb* "*countRows*" justo después de realizar la consulta, y ella nos devuelve el numero de resultados de la última consulta, que se guarda en una variable de sesión.

Puesto que en la consulta solo nos interesa mostrar un número determinado de escuelas, y las demás serán cargadas por ajax para aumentar la velocidad de carga de la pagina, en la consulta *MySql* utilizamos unos límites que serán utilizados también en los archivos con los que trabaja el *ajax* para ir mostrando las escuelas de 9 en 9.

```
$_SESSION['empieza']=0;
$_SESSION['limite']=9;
```

Cada vez que realizamos una consulta cambiamos los limites que son guardados en variables de sesión para poder ser utilizado en otros archivos.

```
$_SESSION['ref']=$sw->pageVars['ref'];
$_SESSION['type']=$sw->pageVars['type'];
$_SESSION['empieza']=$_SESSION['limite'];
$_SESSION['ultima']=$_SESSION['limite']+$_SESSION['limite'];
$_SESSION['restante']=$_SESSION['rows']-$_SESSION['limite'];
```

Después de esta consulta mediante un *foreach()* se comienza a trabajar con los datos de cada una de las escuelas que hayamos obtenido, en primer lugar generamos los marcadores del mapa con los datos obtenidos.

```

$data=$sw->sqlToArray($consulta);    # return a 2 dimensions array
log_msg($data,"THES SCHOOLS JOIN DATA");

foreach($data as $K=>$arr)
{
$lat=$sw->goFast("schools_data","id",$arr['id'], "add_lat");
$lon=$sw->goFast("schools_data","id",$arr['id'], "add_lon");
$rating=$sw->goFast("schools_data","id",$arr['id'], "rating");
$total_rating=$total_rating+$rating;
$rating=($rating*75)/5;

/*ADD MARCADORES*/
$mapa.="
//MARCADOR
var myLatLng = new google.maps.LatLng(".$lat.", ".$lon.");

var infowindow".$numb." = new google.maps.InfoWindow({
content: ".$arr['school_name']."});

var marker".$numb." = new google.maps.Marker({
position: myLatLng,
map: map,
title: ".$arr['school_name']."
});

google.maps.event.addListener(marker".$numb.", 'click', function() {
infowindow".$numb.".open(map,marker".$numb.");
});
";
$numb=$numb+1;

```

Además dentro de este foreach() también se comienza a trabajar y asignar los datos de las escuelas que queremos mostrar, como en el caso de la home, los resultados obtenidos son mostrados también en cajas, como se ha visto también en el diseño, y para ello utilizamos la misma función de *xTemplate* con la que generamos una iteración en el código *htm/* que será mostrado.

```

$xtpl->assign('school_name',$arr['school_name']);
log_msg($arr['school_name'], "THES SCHOOL NAME");
$xtpl->assign('school_image',$site.'/images/escuelas/'.$arr['images_folder'].'/master.jpg');

```

Entre otras cosas como podemos ver en la captura, se asignan al objeto *\$xtp/* el nombre de la escuela, la foto principal, etc.

Por último se genera también dinámicamente las opciones de los filtros del buscador que nos permitirán seleccionar entre diferentes opciones como tipos de curso, alojamientos, duración, fecha de comienzo, etc. Todas estas opciones llamarán a funciones en *jQuery* que recargaran contenido en la pagina mediante *Ajax*.

```
637     $htmlacctypes="";
638     $acctypes=$sw->readTable("accommodation_types_translated");
639
640     foreach($acctypes as $w=>$r)
641     {
642         $htmlacctypes.='<div class="checkbox">
643 <label>
644 <input id="orderBox1" onclick="soloCurso(0);" class="acc" type="checkbox" name="orderBox[]" value="'. $r['id'].'" />
645 '.ucfirst($r['acc_tp'].'_SESSION['ui_lang'].')'.
646 </label>
647 </div>';
648     }
649 }
650 $xtpl->assign('acc_types_html',$htmlacctypes);
651
652
653 $xtpl->assign('cur_types_html',$htmlcurtypes);
654
```

En esta captura de pantalla podemos ver por ejemplo como se genera la lista de tipos de alojamientos que formara parte de una serie de checkbox dentro de los filtros, por último se asignan los tipos generados al objeto \$xtpl para ser llamados desde HTML, también se hace lo mismo con los tipos de curso que son generados de la misma manera.

```
<div class="form-group">
<label for="">{tipo_curso} <a onClick="_gaq.push(['_trackEvent', 'infocourses', 'masinfocourses', 'informacion de cursos ciudad o pais'])"
{cur_types_html}
</div>

<div id="checkboxacc" style="display:none">
<label for="">{tipo_alojamiento}</label>
{acc_types_html}
```

Se puede ver como son llamados luego desde el código html para mostrar el contenido que le hemos asignado.

Por último son también obtenidos otros datos como descripción del destino, y puntos de interés, que se tratan únicamente de textos informativos de cara al usuario de cada destino en particular. Todos estos datos son obtenidos en la misma consulta y asignados de la misma manera para ser mostrados luego en el html.

4.5 Escuela

La página de escuela está manejada directamente por el archivo `school-tpl.php` en el cual como en las anteriores se comienza cargando la plantilla, los módulos que serán necesarios y asignando las palabras y frases que se mostrarán.

```
3  /*TEMPLATE*/
4  $xtpl = new XTemplate('./content/templates/school.tpl'); //Plantilla
5  log_msg('Template Cargado...');
6
7  /*config*/
8  $xtpl->assign('SITE_URL', $sw->coreOptions['site_url']);
9  $site=$sw->coreOptions['site_url'];
10 $lang=$sw->uiLang;
11
12 /*MENU*/
13 $xtpl->assign('menu_contact', ucfirst($sw->uiLang['menu_contact']));
14
15 /*lang engine*/
16 $xtpl->assign('lang_engine', $sw->lang_module);
17 $xtpl->assign('hreflang', $sw->hreflang);
18 $xtpl->assign('this_lang', ucfirst($sw->this_lang));
19
20 /*lang menu*/
21 $xtpl->assign('lang_menu', $sw->lang_menu);
22
23 //Modulo Meta Tags
24 include('./modules/meta_tags.php');
25
26 /*Home URL*/
27
28 $home_url=$sw->readRow("site_urls", array("type"=>"home", "lang_id"=>$sw->goFast("languages", "lang", $_SESSION['ui_lang'])));
29 log_msg($home_url, "HOME URL");
30 $xtpl->assign('home_url', $site.$home_url['full_uri']);
31
32 /*contact URL*/
33
34 $contact_url=$sw->readRow("site_urls", array("type"=>"contact", "lang_id"=>$sw->goFast("languages", "lang", $_SESSION['ui_lang'])));
35 log_msg($contact_url, "HOME URL");
36 $xtpl->assign('contact_url', $site.$contact_url['full_uri']);
37
```

```
37 /*Comenzamos*/
38 $xtpl->assign('title', ucfirst($sw->uiLang['sub_title']));
39 $xtpl->assign('sub_title', ucfirst($sw->uiLang['sub_title2']));
40 $xtpl->assign('destacadas', ucfirst($sw->uiLang['destacadas']));
41 $xtpl->assign('from', $sw->uiLang['from']);
42 $xtpl->assign('menu_contact', $sw->uiLang['menu_contact']);
43 $xtpl->assign('menu_idioma', $sw->uiLang['menu_idioma']);
44 $xtpl->assign('home1', strtoupper($sw->uiLang['home1']));
45 $xtpl->assign('home2', strtoupper($sw->uiLang['home2']));
46 $xtpl->assign('home3', strtoupper($sw->uiLang['home3']));
47 $xtpl->assign('lessons_x_w', ucfirst($sw->uiLang['booking_tool_lessons_x_w']));
48 $xtpl->assign('weeks', $sw->uiLang['booking_tool_course_weeks']);
49 $xtpl->assign('more', $sw->uiLang['booking_tool_more_info']);
50 $xtpl->assign('seleccionacurso', ucfirst($sw->uiLang['inscription_form_select_course']));
51 $xtpl->assign('seleccionaacc', ucfirst($sw->uiLang['inscription_form_select_acc']));
52 $xtpl->assign('resumen', ucfirst($sw->uiLang['booking_tool_summary']));
53 $xtpl->assign('titulo_reserva', ucfirst($sw->uiLang['booking_now']));
54 $xtpl->assign('fecha_inicio', ucfirst($sw->uiLang['search_start_date']));
55 $xtpl->assign('duracion', ucfirst($sw->uiLang['search_duration']));
56 $xtpl->assign('cursoest', ucfirst($sw->uiLang['search_courses']));
57 $xtpl->assign('preciot', ucfirst($sw->uiLang['booking_tool_total_price']));
58 $xtpl->assign('precios', ucfirst($sw->uiLang['booking_tool_price_title']));
59 $xtpl->assign('necesitoacc', ucfirst($sw->uiLang['necesitoacc']));
60 $xtpl->assign('reservarpor', ucfirst($sw->uiLang['booking_tool_stu_x_room']));
61 $xtpl->assign('regimen', ucfirst($sw->uiLang['booking_tool_board']));
62 $xtpl->assign('booknow', ucfirst($sw->uiLang['booking_tool_book_now_b']));
63 $xtpl->assign('tipocurso', ucfirst($sw->uiLang['search_course_type']));
64 $xtpl->assign('tipodiamiento', ucfirst($sw->uiLang['search_accomodation_type']));
65 $xtpl->assign('reservadesde', ucfirst($sw->uiLang['booking_tool_book_now_r']));
66 $xtpl->assign('title_courses', ucfirst($sw->uiLang['search_courses']));
67 $xtpl->assign('primerdia', ucfirst($sw->uiLang['inscription_form_day_start'] . " & " . $sw->uiLang['booking_tool_lesson_duration']));
```

A continuación lo que se hace es obtener desde la base de datos los comentarios de estudiantes para generar el contenido del *Tab* que cuenta con los comentarios, siempre y cuando la escuela en concreto tenga comentarios. Esto se hace mediante una consulta a la base de datos con la *ref* de la pagina, que es el id de la escuela en concreto.

```

78 /*Comentarios*/
79 $fields=array("school_id"=>$sw->pageVars['ref'], "status"=>"1");
80 $comments=$sw->readTable("students_comments", $fields);
81
82 log_msg($comments, 'Comentarios');
83
84 $comments_html='';
85 $link_comments='';
86 if($comments!=NULL){
87
88 $link_comments='<a class="">a href="#comentarios" data-toggle="tab">'.ucfirst($sw->uiLang['opinion']).'</a></a>';
89 $comments_html.='
90 <div id="comentarios" class="tab-pane fade ">
91
92     <div class="col-md-12 school-description">
93         <div class="col-xs-12 col-md-12" id="comentarios">
94
95 //foreach
96
97 foreach($comments as $l=>$com)
98     {
99 $comments_html.='<div class="col-md-12 comments" id="info-comments">
100     <div class="pull-left">
101
102     <p>
103     </div>
104
105     <div class="col-md-12 text-comment" style="margin-top:10px;" id="text-comment">
106     <p>' .ucfirst($com['comment']).'</p>
107

```

Con ello se genera el código *html* que será asignado al objeto *\$xtpl* para ser incluido en el *html*.

También se genera de la misma manera los textos y los iconos que se muestran a la derecha de la pagina y que son informativos y de refuerzo.

```

131 /*FRASES*/
132 $home_txt1=$sw->goFast("content_home", "id", "1", "study ".$_SESSION['ui_lang']);
133 $home_txt2=$sw->goFast("content_home", "id", "1", "safe ".$_SESSION['ui_lang']);
134 $home_txt3=$sw->goFast("content_home", "id", "1", "travel ".$_SESSION['ui_lang']);
135 //log_msg($home_txt1, "Home Text 1");
136 $xtpl->assign('home_text1', ucfirst($home_txt1));
137 $xtpl->assign('home_text2', ucfirst($home_txt2));
138 $xtpl->assign('home_text3', ucfirst($home_txt3));
139
140 if($_SESSION['ui_lang']=="es"){ $uno='Ejercicio Mínimo Garantizado'; $dos=' 7% de descuento'; $tres='Curso online de regalo'; $cuatro='Matrícula incluida'; $cinco='
141 if($_SESSION['ui_lang']=="en"){ $uno='Minimum price guaranteed'; $dos=' 7% discount'; $tres='Free online course'; $cuatro='Registration fee included'; $cinco='Nc
142 if($_SESSION['ui_lang']=="it"){ $uno='Prezzo minimo garantito'; $dos=' 7% di sconto'; $tres='Corso online gratuito'; $cuatro='Quota d'iscrizione inclusa'; $cinco='Nc
143 if($_SESSION['ui_lang']=="de"){ $uno='Minimum price guaranteed'; $dos=' 7% discount'; $tres='Free online course'; $cuatro='Registration fee included'; $cinco='Nc
144 if($_SESSION['ui_lang']=="fr"){ $uno='Minimum price guaranteed'; $dos=' 7% discount'; $tres='Free online course'; $cuatro='Registration fee included'; $cinco='Nc
145 if($_SESSION['ui_lang']=="es"){ $uno='Minimum price guaranteed'; $dos=' 7% discount'; $tres='Free online course'; $cuatro='Registration fee included'; $cinco='Nc
146
147
148
149 $puntosfuertes='<div class="col-xs-12 col-md-12" id="puntosfuertes" style="margin-top: 25px; padding-right: 4px; padding-left: 4px; padding-bottom: 5px;">
150     <h4 class="nomargin" style="font-size: 17px; font-weight: 400;"><span class="glyphicon" style="margin: 3px;"><img class="" style="height
151     <h4 class="nomargin" style="font-size: 17px; font-weight: 400;"><span class="glyphicon" style="margin: 3px;"><img class="" style="height
152     <h4 class="nomargin" style="font-size: 17px; font-weight: 400;"><span class="glyphicon" style="margin: 3px;"><img class="" style="height
153     <h4 class="nomargin" style="font-size: 17px; font-weight: 400;"><span class="glyphicon" style="margin: 3px;"><img class="" style="height
154     <h4 class="nomargin" style="font-size: 17px; font-weight: 400;"><span class="glyphicon" style="margin: 3px;"><img class="" style="height
155     </div>';
156
157 $xtpl->assign('puntosfuertes', $puntosfuertes);
158

```


Por último se añade el modulo de *booking tool* del que ya hemos hablado antes y se encarga de montar todo el sistema de elección y carro de la compra de la aplicación, además de obtener otros datos de la escuela. Si es cierto que hay tareas que realiza el *booking tool* que se pueden sacar y llevar a este archivo de *school-tpl.php*

```
require_once(MAIN_PATH.MODULES_PATH.'booking_tool/booking_tool.php');
```

4.6 Formulario de inscripción

Siguiendo el orden normal de un proceso de compra mediante la web, la siguiente pagina es la pagina de formulario de registro, en la que básicamente lo que se hace es generar un formulario de datos, donde el usuario ha de dejar todos los datos del alumno que se va a matricular en el curso, además se muestra también un resumen de la compra y es la pagina encargada de conectar con los métodos de pago.

Como en las anteriores paginas lo primero que se hace es cargar la plantilla y los modules que harán falta en esta página.

```
3 //ENGINE INSCRIPTION
4
5 $xtpl = new XTemplate('./content/templates/inscription.xtpl'); //Plantilla
6 log_msg($_SESSION,'SESSION');
7 $site=$sw->coreOptions['site_url'];
8 $xtpl->assign('SITE_URL',$sw->coreOptions['site_url']);
9
10 $thanks_url=$sw->readRow('site_urls',array('type'=>'thanks','lang_id'=>$_SESSION['ui_lang_id']));
11 log_msg($thanks_url,'Thanks_URL');
12 /*config*/
13
14 /*MENU*/
15 $xtpl->assign('menu_contact',ucfirst($sw->uiLang['menu_contact']));
16 /*lang engine*/
17 $xtpl->assign('lang_engine',$sw->lang_module);
18 $xtpl->assign('hreflang',$sw->hreflang);
19 $xtpl->assign('this_lang',ucfirst($sw->this_lang));
20 /*lang menu*/
21 $xtpl->assign('lang_menu',$sw->lang_menu);
22 //Modulo Meta Tags
23 include('./modules/meta_tags.php');
24
25 /*Home URL*/
26
27 $home_url=$sw->readRow("site_urls",array("type"=>"home","lang_id"=>$sw->goFast("languages","lang",$_SESSION['ui_lang'])));
28 log_msg($home_url,"HOME URL");
29 $xtpl->assign('home_url',$site.$home_url['full_uri']);
```

En este caso la plantilla utilizada es *inscription.xtpl*. Podemos observar en la captura también que en la línea 6 se hace un *log_msg()* de la sesión, esto es para que se pueda observar en el *debugger* que es lo que contiene la sesión al llegar a este punto, y con ello controlar que se ha generado bien el carro con todos los datos del pedido en la página anterior.

Como en los casos anteriores se aprovecha también el archivo de funciones del modulo de *booking tool* para incluirlo y utilizar las funciones definidas en el en caso de que sea necesario.

Se asignan también todas las palabras o frases que serán visualizadas, utilizando

```
140 //Lang
141 $xtpl->assign ('text_insurance','<a href="http://www.quality-courses.com/insurance_data/' .$_SESSION['ui_lang'] .'.pdf"
142 $xtpl->assign ('text_news',$sw->uiLang['inscription_form_newsletter']);
143 $xtpl->assign ('pagoseguro',$sw->uiLang['pagoseguro']);
144 $xtpl->assign ('text_condi',$sw->uiLang['inscription_form_conditions']);
145 // $xtpl->assign ('forma',ucwords($sw->uiLang['inscription_form_forma']));
146 $xtpl->assign ('enviar',ucfirst($sw->uiLang['contact_form_submit']));
147 $xtpl->assign ('nombres',ucfirst($sw->uiLang['contact_form_firstname']));
148 $xtpl->assign ('apellidos',ucfirst($sw->uiLang['contact_form_lastname']));
149 $xtpl->assign ('email_form',ucfirst($sw->uiLang['contact_form_email']));
150
151 $xtpl->assign ('nacio',ucfirst($sw->uiLang['contact_form_nationality']));
152
153 $xtpl->assign ('level',ucfirst(LANG_LEVEL_L));
154 $xtpl->assign ('male',ucfirst(MALE));
155 $xtpl->assign ('female',ucfirst(FEMALE));
156 $xtpl->assign ('confirma',ucfirst($sw->uiLang['inscription_form_confirm_email']));
157
158 $xtpl->assign ('cumu',ucfirst($sw->uiLang['inscription_form_birthdate']));
159 $xtpl->assign ('telef',ucfirst($sw->uiLang['inscription_form_telephone']));
160 $xtpl->assign ('pass',ucfirst($sw->uiLang['inscription_form_passport']));
161 $xtpl->assign ('genero',ucfirst($sw->uiLang['inscription_form_gender']));
162
163 $xtpl->assign ('coment',ucfirst($sw->uiLang['inscription_form_comments']));
164 $xtpl->assign ('modo',ucfirst($sw->uiLang['inscription_form_payment']));
165
166 $xtpl->assign ('comienzo',ucfirst($sw->uiLang['inscription_form_day_start']));
167 $xtpl->assign ('semanas',ucfirst($sw->uiLang['inscription_form_number_weeks']));
168 $xtpl->assign ('alumnos',ucfirst($sw->uiLang['inscription_form_number_students']));
169 $xtpl->assign ('total',ucfirst($sw->uiLang['inscription_form_total']));
170 $xtpl->assign ('matricula',ucfirst($sw->uiLang['inscription_form_enrollment']));
```

el diccionario cargado en el objeto según el idioma en que se encuentre el usuario.

Uno de los aspectos importantes en este archivo también es a la hora de trabajar con las fechas que tenemos guardadas en la sesión , y que son las fechas escogidas por el usuario para el inicio del curso. Para controlar que los formatos estén correctos se utiliza la función *format_date* que está incluida en el archivo de

```
if($_SESSION['ui_lang']!="en"){ $_SESSION['carro'][$course_std]=format_date($_SESSION['carro'][$course_std],"/","es"); }
```

funciones del modulo *booking tool* que también a sido utilizada con anterioridad.

```
207 //VARIABLES SESSION CARRO
208 log_msg($_SESSION['carro'],'SESSION CARRO');
209 log_msg($_SESSION['carro_acc'],'SESSION CARRO ACC');
210 log_msg($_SESSION['compra_total'],'SESSION TOTAL');
211 //log_msg($_GET['start_day'],'START DAY');
212 //log_msg($_GET['duration_course'],'DURATION');
213
214 $terminos_url=$sw->readRow("site_urls",array("type"=>"terms","lang_id"=>$sw->goFast("languages","lang",$_SESSION['ui_lang']));
215 $xtpl->assign ('terminosurl',$site.$terminos_url['full_uri']);
216 $course_data=$sw->readRow("courses_data","id:".$_SESSION['carro'][$course_id]);
217 log_msg($course_data,'DATOS CURSO');
218 log_msg($_SESSION,'SESSION UI');
219
220 $course_type=$sw->goFast("course_types_translated","id",$course_data['course_type'],'course_tp_'.$_SESSION['ui_lang']);
221 $course_title=ucwords($course_type.' '.$course_data['lessons_x_week'].' '.$sw->uiLang['booking_tool_lessons_x_w']);
222 $xtpl->assign('titulo_curso',ucfirst($course_title));
223
224 // $course_title=$course_type;
225 log_msg($course_title,'Titulo CURSO');
226
227
228 $school_data=$sw->readRow("schools_data","id:".$_SESSION['carro'][$school_id]);
229 log_msg($school_data,'DATOS ESCUELA');
230
231 $school_name=$school_data['school_name'];
232 $xtpl->assign('nombre_escuela',$school_name);
233
234 //Detalles de la reserva-CURSO
```

Como ya se ha comentado esta página se encarga de sacar un resumen del pedido del cliente, para ello se obtienen los datos de las variables de sesión del carro y con ellos se hacen una serie de consultas a la base de datos para sacar todos los detalles del pedido y mostrarlos en el resumen de compra.

Podemos observar como se hace primero un *log_msg()* de las variables de carro, para ver su contenido y a partir de aquí trabajar con ellas. Entre otras cosas obtenemos por ejemplo los detalles de la escuela y el curso seleccionados.

```

350 //Nacionalidades
351 $world_countries=$sw->sqlToArray("SELECT `id`,`country_`.$_SESSION['ui_lang'].'" FROM world_countries ORDER BY `country_`.$_SESSION['ui_lang'].'"
352 // $country_vals=array('=>'Select');
353 $zz=0;
354 $nacionalidades.='<option>'.ucfirst($sw->uiLang['seleccionar']).'</option>';
355 foreach($world_countries as $k=>$arr){
356     if($zz==0){
357         $country_vals[$arr['id']]=$arr['country_`.$_SESSION['ui_lang']'];
358         $nacionalidades.='<option>'. $country_vals[$arr['id']].'</option>';
359         $zz=1;
360     }
361     else
362     {
363         $country_vals[$arr['id']]=$arr['country_`.$_SESSION['ui_lang']'];
364         $nacionalidades.='<option>'. $country_vals[$arr['id']].'</option>';
365     }
366 }
367 $xtpl->assign('nacionalidades',$nacionalidades);
368 //NIVEL
369 $nivel='<option value="" selected="1">'.ucfirst($sw->uiLang['seleccionar']).'</option>
370 <option value="1">'.BEGINNER.'</option>
371 <option value="2">'.LOW.'</option>
372 <option value="3">'.INTERMEDIATE.'</option>
373 <option value="4">'.ADVANCED.'</option>';
374 $xtpl->assign('nivel',$nivel);

```

```

310 //ids de los campos de formulario
311 $firstname="firstname_". $contador;
312 $xtpl->assign('firstname',$firstname);
313 $lastname="lastname_". $contador;
314 $xtpl->assign('lastname',$lastname);
315 $email="email_". $contador;
316 $xtpl->assign('email',$email);
317 $email1="email_1_". $contador;
318 $xtpl->assign('email_1',$email1);
319 $nationality="nationality_". $contador;
320 $xtpl->assign('nationality',$nationality);
321 $birthdate="birthdate_". $contador;
322 $xtpl->assign('birthdate',$birthdate);
323 $lang_level="lang_level_". $contador;
324 $xtpl->assign('lang_level',$lang_level);
325 $gender="gender_". $contador;
326 $xtpl->assign('gender',$gender);
327 $telephone="telephone_". $contador;
328 $xtpl->assign('telephone',$telephone);
329 $passport="passport_". $contador;
330 $xtpl->assign('passport',$passport);
331 $insurance="insurance_". $contador;
332 $xtpl->assign('insurance',$insurance);
333 $newsletter="newsletter_". $contador;
334 $xtpl->assign('newsletter',$newsletter);
335 $comments="comments";
336 $xtpl->assign('comments',$comments);
337

```

Por último se generan dinámicamente los *ids* de los campos del formulario y también algunos campos como por ejemplo el *select* de países de la nacionalidad o nivel de idioma, el *checkbox* con el enlace de los términos y condiciones o el seguro.

Podemos observar que las nacionalidades son sacadas de una consulta en la base de datos donde tenemos una tabla con todos los países del mundo.

En cuanto a las opciones de nivel, utilizamos unas constantes que están definidas en los diccionarios para poner cada nivel en el idioma correspondiente al que nos encontremos.

Los *ids* se generan dinámicamente pensando en que en un futuro podrán ser mas de un alumno a la vez los que se matriculen. Pero de momento la inscripción es simplemente para un alumno.

Dependiendo del método de pago seleccionado el formulario es enviado a una url u otra. Por ejemplo si el método de pago es Transferencia Bancaria, el formulario se envía directamente al archivo checkout.php del que hablaremos a continuación, en cambio si el método de pago es Tarjeta de Crédito, pasamos el formulario a una url donde se crea la conexión con el TPV Virtual del banco y en la cual informamos al usuario de este proceso. Una vez realizado el pago, si el resultado es correcto el encargado de generar la confirmación es el archivo ok.php

4.7 Chackout.php y ok.php

Son los archivos encargados de generar las confirmaciones de inscripción de los clientes. Su funcionamiento es muy similar, la única diferencia es que checkout.php trabaja cuando el método de pago es transferencia bancaria y coge directamente los datos enviados por POST del formulario de inscripción y genera el pedido, lo guarda en la base de datos y genera también dos emails, uno para el cliente para confirmar su pedido y otro para la empresa en el cual se detallan los datos de la compra. Si el método de pago es por tarjeta de crédito también se llevan los datos al checkout.php pero lo que se hace es que se establece la conexión con el TPV del banco y se informa al usuario de que será redirigido a él para un pago más seguro.

```
$thanks_url=$sw->readRow('site_urls',array('type'=>'thanks','lang_id'=>$_SESSION['ui_lang_id']));
//CHECKOUT
if ($_POST['payment_method']=='credit_card'){

    $m_ko=trim($sw->coreOptions['site_url']).'/ko.php';
    $m_ok=trim($sw->coreOptions['site_url']).'/ok.php';

    $m_commerce_name="QUALITY CHOICES WORLDWIDE";

    $sis_url_test="https://sis-t.sermepa.es:25443/sis/realizarPago"; # url de prueba
    $sis_url_real="https://sis.sermepa.es/sis/realizarPago"; # url entorno real

    $sis_url=$sis_url_real;

    $m_key_test='qwertyasdf0123456789';
    $m_key_real='S6766072S1505802';

    $m_key=$m_key_real;

    $m_transaction_type='O';

    $m_return_url=trim($sw->coreOptions['site_url']).'/checkout.php';

    $m_code='285468732';
    $m_terminal='2';
```

Se puede observar que una de las primeras cosas que se realizan es obtener la url de la pagina de gracias en el idioma en que nos encontremos y después se

comprueba mediante un if si el método de pago es tarjeta de crédito. Si es este el caso se comienza a configurar las variables con las opciones dadas por el banco para la conexión con su pasarela de pago y se genera un formulario con todos

```
$consumer_lang=$sw->goFast('languages','id',$SESSION['ui_lang_id'],'tpv_code');
if(!$consumer_lang){ $consumer_lang=$sw->goFast('languages','id',1,'tpv_code'); }

$form='
<link rel="stylesheet" href="http://quality-courses.com/test/content/templates/bootstrap.css" media="screen">
<link rel="stylesheet" href="http://quality-courses.com/test/content/templates/css/bootswatch.min.css">
<div id="paso_tpv" class="col-md-12" style="margin-left: auto; margin-right: auto; text-align: center;">
<form id="tpv_vars" action="',$sis_url.'" method="POST">

<input type="hidden" name="total_payment" id="total_payment" style="margin:0px;padding:3px;border:1px inset;" value="',$amount.'">
<input type="hidden" name="Ds_Merchant_Amount" id="merchant_amount" style="margin:0px;padding:3px;border:1px inset;" value="',$amount.'">
<input type="hidden" name="Ds_Merchant_Currency" id="pcc-gen139" style="margin:0px;padding:3px;border:1px inset;" value="',$currency.'">
<input type="hidden" name="Ds_Merchant_Order" id="merchant_order" style="margin:0px;padding:3px;border:1px inset;" value="',$order.'">
<input type="hidden" value="',$m_code.'" name="Ds_Merchant_MerchantCode" id="pcc-gen148" style="margin:0px;padding:3px;border:1px inset;">
<input type="hidden" name="Ds_Merchant_TransactionType" id="pcc-gen149" style="margin:0px;padding:3px;border:1px inset;" value="',$m_transaction.'">
<input type="hidden" name="Ds_Merchant_Terminal" id="pcc-gen154" style="margin:0px;padding:3px;border:1px inset;" value="',$m_terminal.'">
<input type="hidden" name="Ds_Merchant_MerchantURL" id="pcc-gen159" style="margin:0px;padding:3px;border:1px inset;" value="',$m_return_url.'">
<input type="hidden" name="Ds_Merchant_MerchantSignature" id="merchant_signature" style="margin:0px;padding:3px;border:1px inset;" value="',$sig.'">
<input type="hidden" name="Ds_Merchant_ConsumerLanguage" id="pcc-gen168" style="margin:0px;padding:3px;border:1px inset;" value="',$consumer_lang.'">
<input type="hidden" name="Ds_Merchant_MerchantName" id="pcc-gen173" style="margin:0px;padding:3px;border:1px inset;" value="',$m_commerce_name.'">
<input type="hidden" name="Ds_Merchant_UrlOK" id="pcc-gen178" style="margin:0px;padding:3px;border:1px inset;" value="',$m_ok.'">
<input type="hidden" name="Ds_Merchant_MerchantData" id="pcc-gen125" style="margin:0px;padding:3px;border:1px inset;" value="',$SESSION['unique

<h2>',$text.'"</h2>
<input class="btn btn-warning" type="submit" value="',$ucfirst($lang['btn-pagar']).'">
```

estos datos, además de informar al usuario a donde será redirigido para un pago más seguro.

Desde aquí si el pago es correcto se pasa a la pagina de ok.php. En caso de que el pago sea transferencia bancaria se comienzan a preparar los campos para insertarlos en la base de datos.

```
if($_POST['payment_method']=='bank_transfer')
{
    //Datos
    $tpv_subject='';
    $fields=array();
    $course_id=$SESSION['carro']['course_id'];
    $course_data=$sw->readRow('courses_data','id',$course_id);
    $city_id=$sw->goFast('schools_data','id',$SESSION['course_bt']['school_id'],'city_id');
    $city_data=$sw->readRow('city_names','id',$city_id);

    //Numero de Alumnos
    $alumnos=$SESSION['carro']['alumnos'];
    //echo "hasta aqui";
    $i=0;
    //while($i<=$alumnos)
    //{
        $fields['random']=uniqid(rand(),true);

        $fields['site']='www.quality-courses.com';
        $fields['language']=$SESSION['ui_lang'];

        //si i=0 alumno 1
        //Datos Alumnos 1
        $fields['firstname']=$_POST['firstname_1'];
        $fields['surname']=$_POST['lastname_1'];
    }
```

Para ello y puesto que los datos de las compras no se guardan en la misma base de datos que utiliza la web, se hace una desconexión de la anterior base de datos y una conexión a la nueva.

```
//Fin conexion
/*NUEVA CONEXION*/

$sw->dbClose();

$sw->dbConnect('localhost','d150210_master','wvsiwvg','d150210_prices2010');
```

```
foreach($final as $inscription)
{
    $country_id=$inscription['country_id'];
    unset($inscription['country_id']);
    $accommodation_title=$inscription['accommodation_title'];
    $accommodation_board=$inscription['accommodation_board'];
    unset($inscription['accommodation_title']);
    unset($inscription['accommodation_board']);
    $airport_name=@$inscription['airport_name'];
    unset($inscription['airport_name']);
    $original_course_price=null;
    $original_course_regfee=null;
    $original_accommodation_price=null;
    $original_acc_fee=null;
    $original_insurance_price=null;
    $original_transfer_price=null;
    $duration=($inscription['duration']>1) ? $inscription['duration'] .' '.WEEKS : $inscription['duration'] .' '.WEEK;
    $inscription['duration']=(($inscription['duration']>1) ? $inscription['duration'] .'weeks' : $inscription['duration'] .'week';
    $inscription['total_ammount']=$fields['total_ammount'];
    /*INSERTAR DATOS*/
    $sw->insertRow('inscriptions_copy',$inscription);
}
```

A continuación se preparan los datos y se guardan en la *db*.

Una vez insertados los datos se pasa a generar los emails que serán enviados tanto al usuario como a la empresa con el resumen del pedido. Para ello se genera el email de la empresa en la variable *\$mail_body* y el del cliente en la variable *\$client_email_body*

```

$mail_body=" ";

$mail_body.="----- " .EM_CLIENT_DATA_T." -----\n\n";
$mail_body.=EM_FIRSNOME_L." : ".ucwords($inscription['firstname'])."\n";

$mail_body.=EM_SURNAME_L." : ".ucwords($inscription['surname'])."\n";

$mail_body.=EM_SEX_L." : ".transGender($inscription['gender'])."\n";

$mail_body.=EM_BIRTHDATE_L." : ".format_date($fields['age'],'/',$SESSION['ui_lang'])."\n";
$mail_body.=EM_CLIENT_EMAIL_L." : ".$inscription['email']."\n";
$mail_body.=EM_PASSPORT_L." : ".$inscription['passport']."\n";
$mail_body.=EM_TEL_L." : ".$inscription['telephone']."\n";

$mail_body.="
----- " .EM_COURSE_DATA_T." -----\n\n";
$mail_body.=EM_COURSE_CITY_L." : ".$school_name."\n";
$mail_body.=EM_COURSE_TYPE_L." : ".$inscription['course']." ".LPW."\n";

$mail_body.=EM_COURSE_DURATION_L." : ".$duration."\n";

$mail_body.=EM_COURSE_SD_L." : ".format_date($inscription['startingdate'],'/',$SESSION['ui_lang'])."\n";

$mail_body.=LANG_LEVEL_L." : ".lang_level($inscription['languageLevel'])."\n";

```

Y son enviados mediante la función mail de php, una vez enviados se redirige la pagina hacia la pagina de gracias.

```

$mail_headers="From: ".$inscription['email']."\r\n";
$mail_headers.="form-type: inscription\r\n";
$mail_headers.="record-id: ".$last_insert_id."\r\n";
mail('info@quality-courses.com',$mail_subject,$mail_body,$mail_headers);

$client_mail_headers="From: info@quality-courses.com\r\n";
$client_mail_headers.='MIME-Version: 1.0' . "\r\n";
$client_mail_headers.='Content-type: text/html; charset=iso-8859-1' . "\r\n";
$client_email_body=str_replace('{firstname}',$inscription['firstname'],CLIENT_EM_HEAD_TEXT);
mail($inscription['email'],CLIENT_EMAIL_SUBJECT,$client_email_body,$client_mail_headers);
echo("<script>location.href = 'http://www.quality-courses.com' . $thanks_url['full_uri'] . ' . '</script>");

```

En cambio el archivo ok.php entra en funcionamiento cuando el método de pago es tarjeta de crédito y el pago se ha realizado con éxito, obtiene los datos del pedido y del formulario de inscripción desde la variable de sesión y para ello en el paso intermedio en que se informa al usuario de que será trasladado a la pasarela de pago del banco se guarda una sesión con todos los datos del *POST* del formulario de inscripción. Y con estos datos se encarga de generar el registro para la base de datos y también los emails, como en el caso anterior para el cliente y para la empresa.

Este proceso es exactamente igual que en la página de checkout.php la única diferencia como hemos comentado es que los datos del *\$_POST* vienen por variable de sesión.

4.8 Contacto

La pagina de contacto cuenta con una estructura de programación sencilla pero es una página importante para la empresa, ya que es uno de los medios de comunicación con los clientes.

El archivo php que se encarga de gestionarla es contact-tpl.php y como en todos los casos anteriores lo primero que hace es cargar la plantilla y los módulos que puedan ser necesarios. Y después se comienza a asignar los textos que aparecerán en la pagina desde los diccionarios.

```
$xtpl = new XTemplate('./content/templates/contact.xtpl'); //Plantilla
log_msg($_SESSION, 'SESSION');
$site=$sw->coreOptions['site_url'];
$xtpl->assign('SITE_URL', $sw->coreOptions['site_url']);

$thanks_url=$sw->readRow('site_urls', array('type'=>'thanks', 'lang_id'=>$_SESSION['ui_lang_id']));
log_msg($thanks_url, 'Thanks_URL');
/*config*/
//$xtpl->assign('SITE_URL', $sw->coreOptions['site_url']);
//$site=$sw->coreOptions['site_url'];
/*MENU*/
$xtpl->assign('menu_contact', ucfirst($sw->uiLang['menu_contact']));
/*lang engine*/
$xtpl->assign('lang_engine', $sw->lang_module);
$xtpl->assign('hreflang', $sw->hreflang);
$xtpl->assign('this_lang', ucfirst($sw->this_lang));
/*lang menu*/
$xtpl->assign('lang_menu', $sw->lang_menu);
//Modulo Meta Tags
include('./modules/meta_tags.php');

/*Home URL*/

$home_url=$sw->readRow("site_urls", array("type"=>"home", "lang_id"=>$sw->goFast("languages", "lang", $_SESSION['ui_lang'])));
log_msg($home_url, "HOME_URL");
$xtpl->assign('home_url', $site.$home_url['full_uri']);
```

Textos del formulario:

```
$xtpl->assign ('enviar', ucwords($sw->uiLang['contact_form_submit']));
$xtpl->assign ('nombre', ucwords($sw->uiLang['contact_form_firstname']));
$xtpl->assign ('apellidos', ucwords($sw->uiLang['contact_form_lastname']));
$xtpl->assign ('email_form', ucwords($sw->uiLang['contact_form_email']));
$xtpl->assign ('telef', ucwords($sw->uiLang['inscription_form_telephone']));
$xtpl->assign ('direccion', ucwords($sw->uiLang['direccion']));
$xtpl->assign ('donde', ucwords($sw->uiLang['donde']));
```

El formulario envía sus datos a este mismo archivo, y el se encarga de comprobar si están llegando datos por POST, en ese caso crea el email y lo envía mediante la función *mail()* de *php* y muestra un mensaje en pantalla recargando la pagina con la confirmación del envío.


```

if($_POST['nombre']){

$email_address="info@quality-courses.com";
$mail_body="".ucwords($sw->uiLang['contact_form_firstname']).":" .$_POST['nombre']."\r\n";
$mail_body.="".ucwords($sw->uiLang['contact_form_lastname']).":" .$_POST['apellidos']."\r\n";
//$mail_body.="País:" .$_POST['pais']."\r\n";
$mail_body.="".ucwords($sw->uiLang['contact_form_email']).":" .$_POST['email']."\r\n";
$mail_body.="".ucwords($sw->uiLang['inscription_form_telephone']).":" .$_POST['telefono']."\r\n";
$mail_body.="".ucwords($sw->uiLang['contact_message']).":" .$_POST['mensaje']."\r\n";

$mail_subject="INFORMATION FORM QUALITY-COURSES.COM";

    $mail_headers="From: " .$_POST['email']."\r\n";
    $mail_headers.="form-type: contact\r\n";

    mail($email_address,$mail_subject,$mail_body,$mail_headers);
    //mail('info@quality-courses.com',$mail_subject,$mail_body,$mail_headers);

$form_html=ucwords($sw->uiLang['contact_msg']);
$tpl->assign('formulario',$form_html);

```

En caso contrario muestra el formulario de contacto.

Como se puede observar en la captura mediante un *if()* se controla la llegada de datos por POST , en este caso con el \$_POST[nombre] que es uno de los campos del formulario. Y se asigna a la variable *\$form_html* el mensaje de confirmación de envío.

```

)else(

$form_html='<form class="form-horizontal" method="post">
<fieldset>

<!-- Form Name -->
<legend>'.ucwords($sw->uiLang['contact_form_title']).'</legend>

<!-- Text input-->
<div class="form-group col-md-6" style="margin-left:15px;">
<label class="control-label">'.ucwords($sw->uiLang['contact_form_firstname']).':</label>
<div class="controls">
<input tabindex="1" id="nombre" name="nombre" type="text" placeholder="'.ucwords($sw->uiLang['contact_form_firstname']).'" class="col-md-6 form-control"

</div>
</div>

<!-- Text input-->
<div class="form-group col-md-6" style="margin-left:15px;">
<label class="control-label">'.ucwords($sw->uiLang['contact_form_lastname']).':</label>
<div class="controls">
<input tabindex="1" id="apellidos" name="apellidos" type="text" placeholder="'.ucwords($sw->uiLang['contact_form_lastname']).'" class="col-md-6 form-cont

```

En caso de que no llegue ningún dato por *POST* se crea el *html* del formulario en la variable *\$form_html* y esta es luego asignada al objeto *\$xtpl* para ser llamada desde el código *html*.

Todo el contenido que aparece además del formulario y los textos, como por ejemplo el mapa de situación de la empresa, están insertados directamente en el *HTML*.

4.9 Ofertas

Esta es una de las últimas páginas agregadas a la web, en la que se pretende mostrar aquellas escuelas en las cuales se cuente con alguna oferta o promoción especial.

Se controla desde el archivo *offers-tpl.php* y la plantilla que se carga es *offers.xtpl*.

```

$value="1";
$order="ORDER BY `desc` DESC";
$offers_data=$sw->readTable("offers","status:". $value. "", $order);

foreach($offers_data as $s=>$arrS)
{
    $fields=array("id"=>$arrS['id_school'], "status"=>1);
    $schools_data=$sw->readRow("schools_data", $fields);
    //print_r($schools_data);
    $xtpl->assign('school_name', ucfirst($schools_data['school_name']));
    $xtpl->assign('school_image', $site.'/images/escuelas/'.$schools_data['images_folder'].'/master.jpg');
    $city_name=$sw->goFast("cities_translate", "id_city", $schools_data['city_id'], "value_".$_SESSION['ui_lang']);
    if($city_name==NULL){ $city_name=$sw->goFast("city_names", "id", $schools_data['city_id'], "value");}

    $xtpl->assign('city_name', ucfirst($city_name));
    $xtpl->assign('desc', $arrS['desc']);

    $language=$sw->goFast("languages_translated", "id", $schools_data['study_lang'], "name_".$_SESSION['ui_lang']);
    $xtpl->assign('language', $language);
    $xtpl->assign('school_of', ucfirst($sw->uiLang['school_of']));

    $school_url=$sw->readRow("site_urls", array("type"=>"school", "ref"=>$schools_data['id'], "lang_id"=>$sw->goFast("languages", "lang", $_SESSION['ui_lang']));
    $xtpl->assign('school_url', $site.$school_url['full_uri']);
    $city_url_sql="SELECT * FROM site_urls, languages, cities_data WHERE site_urls.type='city' AND cities_data.city_id='$schools_data[city_id]' AND";
    $city_url=$sw->sqlToArray($city_url_sql);
    $xtpl->assign('city_url', $site.$city_url[0]['full_uri']);

    $rating=$schools_data['rating'];
    $rating=($rating*75)/5;

```

Se repite en el mismo proceso que en las paginas anteriores, se inserta la plantilla que se va a usar, se cargan los módulos necesarios, y los textos.

La parte importante de este archivo comienza con la consulta a la base de datos, concretamente a la tabla *offers* donde se encuentra la id de las escuelas y el descuento especial que se tiene en ellas. A partir de aquí se genera el contenido en cajas como las de la pagina de ciudad o país, aunque modificando algunos aspectos estéticos en HTML.

4.10 Páginas default

Las que llamamos paginas default son aquellas que tienen un contenido mas estático, normalmente solo texto, o texto y formulario de contacto. Pero que hacen uso de la pagina php *default-tpl.php* y la plantilla *default.xtpl*.

Estas páginas son por ejemplo el quienes somos, condiciones generales, o paginas con textos como cursos de larga duración o trabajar y estudiar en el extranjero.

Su funcionamiento es más sencillo que los anteriores, para comenzar se carga como en los casos anteriores la plantilla, módulos y algunos textos. A continuación se hace una consulta a la base de datos con el *type* de la *url* y se sacan los datos

```
$sql="SELECT content_'$_SESSION['ui_lang'].'" FROM independent_content WHERE `type`='$_SESSION['pageVars']['type'].'";  
$resultado=$sw->sqlToArray($sql);  
$texto=$resultado[0]['content_'$_SESSION['ui_lang'].''];
```

según el idioma en el que estemos también.

Como podemos ver en la captura se hace una consulta a la tabla *independent_content* seleccionando el contenido según el idioma y el *type* que son datos que nos da la *url*.

En esta tabla se encuentra el contenido en *html* redactado, simplemente se asigna al objeto *\$xtpl* y es mostrado en el *HTML*.

*En el caso de la pagina de trabajar y estudiar o de año escolar en el extranjero que cuentan con un formulario de contacto, se tratan de una variable del default en el que simplemente añadimos el mismo formulario que tenemos en contacto. Los archivos son *work-tpl.php* y *work.xtpl*

4.11 Ajax/jQuery

En muchas de las paginas se utiliza jQuery para cargar contenido en pantalla sin recargar la pagina mediante Ajax. El hecho de que se utilice jQuery para hacer la llamada ajax simplifica mucho el trabajo.

Puesto que todas las llamadas son más o menos similares comentaremos algunas como ejemplo. Una de las paginas que mas ajax utiliza es la pagina de ciudad y país, puesto que casi todas las opciones de los filtros que nos permiten seleccionar las opciones que queremos, cargan mediante *ajax* las escuelas en la zona de resultados.

Lo primero que tenemos que comentar es el código *jQuery* que tenemos en la plantilla *html*/para que haga la llamada al archivo correspondiente.

```
<script>
function solocurso(limite){
    var ui_lang = $("#idioma").val();
    var duration = $("#duration").val();
    var fecha=$("#fecha").val();

    if($("#optionsRadios1").is(':checked')) {
        var alojamiento = 0;
        $("#checkacc").css("display","none");
    } else {
        if($("#optionsRadios2").is(':checked')) {
            var alojamiento = 1;
            $("#checkacc").css("display","");
        }
    }
};

var values = $('input:checkbox:checked.acc').map(function () {
    return this.value;
}).get();

var values2 = $('input:checkbox:checked.cur').map(function () {
    return this.value;
}).get();
```

```

var parametros = {
    "limite" : limite,
    "duration" : duration,
    "alojamiento" : alojamiento,
    "acc" : values,
    "fecha" : fecha,
    "ui_lang" : ui_lang,
    "cur" : values2
};

$.ajax({
    data: parametros,
    url: '{SITE_URL}/solocurso.php',
    type: 'GET',
    beforeSend: function () {
        $("#resultados").html("<img id='loading' style='height:
        $("#primera").css("display", "none");
    },
    success: function (response) {
        $("#primera").css("display", "none");
        $("#loading").remove();
        $('#resultados').html(response);
    }
});
}
</script>

```

Podemos ver en esta captura como mediante *jQuery* primero creamos variables con las que obtenemos los valores de los campos de los filtros. Una vez tenemos todos los datos los guardamos todos dentro de la variable `parámetros` y utilizamos `$.ajax({})` para establecer los parámetros de conexión con el archivo *php* que se encargara de procesar y devolvernos los datos.

En primer lugar asignamos a `data` los datos que queremos enviar, en segundo lugar necesitamos la `url` del archivo al que vamos a llamar, en este caso `solocurso.php`, en tercer lugar se asigna el `type`, que es como serán enviados estos datos, en este caso mediante `GET`.

Y ahora establecemos dos funciones, una para antes del envío, en la cual pintamos en pantalla una imagen de *loading* para informar al usuario de que se esta cargando el contenido. Y la otra función para cuando tenemos la devolución del contenido, en la cual primero se elimina la imagen de *loading* y luego se carga en el div indicado la respuesta del archivo en cuestión.

Para entender mejor que es lo que ocurre en este ejemplo pasaremos a revisar el archivo `solocurso.php` que es el encargado de devolvernos la respuesta del `ajax`.

solocurso.php

Este archivo lo que hace es recibir como hemos visto mediante *Get* en el código de la llamada *ajax* de *jQuery* todos los datos del formulario de filtros en las páginas de ciudad y país.

Lo primero que se hace en este archivo y en los demás que son utilizados con *ajax* es incluir los módulos, el diccionario y el archivo de configuración para poder establecer conexiones con la base de datos.

```
<?PHP
    session_start();
    require_once('./sw-config.php');
    $sw=new SimpleWebsite('',false);
    $sw->magicQuotes();           # check if magic quotes is enabled
    //Conexion a la db y web options
require(MAIN_PATH.'connect.php');
//Include los plugins/modules
//require_once(MAIN_PATH.'sw-plugins.php');
$_SESSION['ui_lang']=$_GET['ui_lang'];
require_once(MAIN_PATH.'modules/booking_tool/functions.php');
require_once(MAIN_PATH.'modules/lang_engine/languages/'.'$_SESSION['ui_lang'].'.php');
$sw->addVars($lang,"uiLang");
$site=$sw->coreOptions['site_url'];
```

Vemos que como se trata de un archivo que esta fuera del *site*, tenemos que volver a inicializar el objeto *\$sw* y volver a incluir los archivos de *sw-config.php* y *connect.php*. Además también se incluye el archivo de funciones del *booking tool* por si tenemos que trabajar con algunas de las funciones allí establecidas y el diccionario del lenguaje en el cual nos encontremos y lo sabemos gracias a la variable de sesión.

Una vez se tiene la configuración establecida, se comienza a guardar en variables los datos que llegan mediante *\$_GET* y a crear los filtros de los que depende para luego hacer las consultas a la base de datos.

Algunos de los datos que este archivo recibe son, duración, fecha de comienzo, si se quiere o no alojamiento, si se quiere filtrar por algún tipo de curso en especifico, o algún tipo de alojamiento...etc.

Con estos datos construimos una serie de filtros que son aplicados luego en caso de ser necesario en las consultas.

```

$duration=$_GET['duration'];

$filtero_acc="";
if(@$_GET['acc']){
    $acc=$_GET['acc'];

    $bandera=0;
    foreach($acc as $ac)
    {
        if($bandera==0){
            $filtro_acc.=" AND (accommodations_data.acc_title='".$.$ac."'";
            $bandera=1;
        }
        else{
            $filtro_acc.=" OR accommodations_data.acc_title='".$.$ac."'";
        }
    }
    $filtro_acc.=")";
}

```

Por ejemplo en este caso creamos los filtros para la consulta por alojamientos, el primer filtro lo creamos con un AND y los siguientes con un OR para cubrir todas las posibilidades si el usuario selecciona más de un tipo de alojamiento en sus filtros.

```

if($_SESSION['type']=="country" && $alojamiento==0){
    $consulta2="SELECT DISTINCT `schools_data`.`id`, `schools_data`.`video`, `schools_data`.`school_description_es`, `schools_data`.`school_address`,
    `schools_data`.`images_folder`, `schools_data`.`main_image`, `schools_data`.`school_name`, `schools_data`.`comision`, `courses_data`.`course_type`,
    `courses_data`.`lessons_x_week`, `course_prices`.`".$duration."` as c_price, `course_prices`.`currency_id`, `course_prices`.`admin_fee`, `cities_da
    JOIN `courses_data` ON `schools_data`.`id` = `courses_data`.`school_id` JOIN `course_prices` ON `courses_data`.`id` = `course_prices`.`course_id`
    JOIN `cities_data` ON `schools_data`.`city_id` = `cities_data`.`city_id` JOIN `city_names` ON `cities_data`.`city_id` = `city_names`.`id` JOIN `cou
    WHERE `countries_data`.`id` = '".$_SESSION['ref']."' AND `schools_data`.`status` = '1' ".$filtro_cur." AND `courses_data`.`status` = '1'
    AND `course_prices`.`".$duration."` > 0.00 ORDER BY `course_prices`.`".$duration."` ASC ";
}

```

Estos filtros son luego utilizados en las consultas, donde tenemos una consulta para cada una de las posibilidades (pagina de país sin alojamiento, pagina de país con alojamiento, pagina de ciudad sin alojamiento y pagina de ciudad con alojamiento) Todo ello se controla mediante *if()*, variables de sesión y los datos recogidos por *\$_GET*

Con estas consultas como en la página de ciudad o país obtenemos todos los datos sobre las escuelas de este destino y sobre los cursos que estamos filtrando. Mediante un *foreach()* de estos datos separamos escuela por escuela y curso por curso y creamos el *html* en la variable *\$html*.

```
$html.= '<div class="col-mq-4">
    <div class="thumbnail caja" style="height: 440px;">
        <a href="'. $site.$school_url['full_uri'].'?course_id='.$sarr['course_id'].'&dur
        
        <span class="label label-success"><div class="content" style="font-size: 14px
        <div class="caption">
            <h3 class="title" style="margin-top:15px; margin-bottom:0px; color:#f1592
            <h4 style="margin-top:0px; font-size: 16px;"><a href="'. $site.$city_url[0]

            <ul class="list-unstyled">
                <li><span><strong>'.ucfirst($course_type).'</strong></span>
                </li>
                <li><span><strong>'.ucfirst($sw->uiLang['booking_tool_lessons_x_w']).'
                </li>
                <li><span><strong>'.ucfirst($acc_type).'</strong></span>
                </li>
            </ul>
            <div class="ratings">
                <!--<p class="pull-right">6 reviews</p-->
                <a href="'. $site.$school_url['full_uri'].'?course_id='.$sarr['course_id'].'
                <span class="ui-rater-starsOff" style="width: 75px; cursor: pointer; opacity: 1;margin-top: 10px;

```

Por último para devolver este contenido a la función de *jQuery* que se encargara de pintarlo en el *div* correspondiente es tan sencillo como realizar un *echo* de esta variable que incluye todo el código *html* de cada uno de los cuadros de escuelas que veremos después en los resultados.

```
echo $html;
if($_SESSION['reservas']>=0){
if($alojamiento==0){
$carga='<div style="text-align:center;"><button style="background-color:#34495E; border-color:#34495E;" align="center" type="button" id="primera" href=""
echo $carga;
if($alojamiento==1){
$carga='<div style="text-align:center;"><button style="background-color:#34495E; border-color:#34495E;" align="center" type="button" id="primera" href=""
echo $carga;
}
```

En esta captura se puede observar también que después del *echo \$html* con el contenido de los resultados se controla si hay más de 9 escuelas para la paginación y se añade también con un *echo* el botón encargado de cargar mas escuelas.

Como se ha comentado ya, todas las funciones que utilizan ajax funcionan de una manera similar y son las siguientes.

"cargaracc.php y cargarcursos.php"

Son los encargados de cargar los datos de alojamientos y cursos desde la pagina de escuela. La función jQuery que los llama es ejecutada cuando el usuario selecciona una fecha de comienzo o una duración en la página de escuela.

"cargarescuelas.php y cargarescuelasacc.php"

Son los encargados de controlar la paginación, son llamados desde las páginas de ciudad o país cuando hay más de 9 resultados (establecido por defecto) y se clica en el botón estos archivos cargan los resultados de 9 en 9. Uno lo hace cuando tenemos seleccionada la opción de solo curso y el otro cuando tenemos marcado el filtro de curso mas alojamiento.

"carro.php"

Por último tenemos el *ajax* del carro, que se encarga de crear todas las variables de sesión del carro de la compra con las selección de los productos que el usuario quiera y además nos devuelve en *html* un resumen de la compra que es cargado en la sección de *booking tool* de la pagina de escuela.

5. Usabilidad.

5.1. Pruebas de usabilidad versión final.

5. Usabilidad

Estas pruebas son realizadas sobre el primer boceto que se probó en el dominio aprenderidiomas.com y sobre la web antigua de *qualitycourses*.

En primer lugar las pruebas que se han realizado sobre unas webs ya operativas (aunque en versión beta), de las cuales tenemos datos y el porcentaje de rebote es alto, lo cual quiere decir que un número elevado de visitas abandona la página sin ni siquiera navegar. La idea principal es hacer un plan de evaluación con una lista de tareas, grabación y aplicar algunos pequeños cuestionarios (pos-test) y focus-group a usuarios reales. Se aplica también *thinking-aloud* para recoger comentarios y observaciones de los usuarios en tiempo real.

Lo primero que se hace es la preparación del test de usabilidad donde se pretende comparar la versión beta con la antigua versión de la web, para encontrar posibles fallos en un lado y en otro y observar también el comportamiento de los usuarios ante dos webs que son muy distintas en cuanto a diseño.

Se realizan algunos posibles escenarios en cada una de las webs que queremos analizar y a cada usuario se le asigna un escenario para que realice las tareas que en él se piden, mientras se van anotando todo lo que ocurre, lo que el usuario dice y posibles problemas de usabilidad que se detecten. También se van a realizar grabaciones de la realización de las tareas por parte de los usuarios y un seguimiento de eye tracking.

En los datos anexos se entregan los modelos de test de usabilidad para cada web, así como también los cuestionarios de usabilidad y del focus group.

Después de la realización de los test de usabilidad los usuarios pasan a una sala donde son atendidos por una persona de la empresa, que les entrega dichos formularios y se charla un poco con ellos sobre algunas cuestiones que creemos relevantes saber para aprender cuáles son los errores que se pueden estar cometiendo o en que se puede mejorar.

Anexo también los cuestionarios de las personas que se prestaron a la muestra.

Pese a que las pruebas son realizadas sobre una muestra pequeña de personas, con todas las anotaciones que se sacan y el resultado de los cuestionarios somos capaces de detectar algunas deficiencias tanto en el nuevo diseño como en el antiguo y también algunas ideas para mejorar de cara al futuro.

Como conclusión de estas pruebas sacamos varios aspectos en los que se cree que fallamos y por ello se da una vuelta al proyecto para realizarlo de nuevo pero esta vez para la web principal de la empresa.

-Menús con demasiadas opciones y que no se utilizan.

-Complicaciones de los usuarios para seleccionar fechas de comienzo y duración de los cursos.

-Falta de información o información no encontrada por los usuarios en cursos y alojamientos.

-Problemas de los usuarios para encontrar algún destino en particular.

Estos patrones de error se repiten tanto en la web antigua como en la nueva beta, como dato positivo podemos sacar que por comentarios de los usuarios la beta gusta más a nivel visual, y daba la impresión de una web más moderna con respecto a la anterior.

Los problemas más graves encontrados se vieron en el formulario de inscripción de la antigua web donde habían demasiados campos y se tardaba demasiado en rellenar, además de que el campo de fecha de nacimiento comenzaba en el día actual y era un problema para los usuarios retroceder hasta sus fechas de nacimiento.

Otro de los problemas graves era la falta de claridad para encontrar destinos, tanto para los usuarios que usaban el menú superior como los que utilizaban el buscador.

En la página de escuela también se detectó un problema a la hora de mostrar detalles sobre cursos, ya que los usuarios no eran capaces de encontrarlos con facilidad.

Al reverso de los cuestionarios de los usuarios que realizaron las pruebas se han anotado los problemas y puntos que fueron detectados en su realización.

A pesar de que se encontraron errores en los cuestionarios de usabilidad los usuarios afirman que las aplicaciones son fáciles de utilizar y no suponen muchos problemas para ellos.

Los datos que se pueden sacar de las cuestiones del *focus group* son menos relacionadas con la aplicación web, y más con las tendencias de los usuarios, primeras impresiones y las imágenes que le da la marca o el producto a través de la web, de la cual también se pueden sacar algunos detalles a los que prestar importancia. Subrayados en cada una de las respuestas de los anexos.

Aunque es un apartado en el que se ha intentado tener especial atención, nos hubiese gustado poder tener un mayor número de pruebas y número de muestras

para obtener aun mas resultados, ya que por falta de recursos y tiempo no se ha podido.

5.1 Pruebas de usabilidad en la version final

En la versión final de la web no se han podido realizar pruebas de usabilidad en laboratorio, pero se ha intentado buscar algunas alternativas, y para ello se ha utilizado el software *crazy-egg*.

Este software hace un análisis del comportamiento de los usuarios en la web, muestra todas las zonas donde clican los usuarios, mapas de calor con las zonas donde se detienen mas los usuarios y también analiza la procedencia de los usuarios, de donde vienen los usuarios que llegan a la web y como se comportan.



▶ SNAPSHOTS □	CREATED □	UPDATED □	STATUS □
✓			<input checked="" type="checkbox"/> RUNNING <input type="checkbox"/> STOPPED
<input type="checkbox"/> Home - QualityCourses	Jun 4, 2014	Today	<input checked="" type="checkbox"/> running
<input type="checkbox"/> Paginas de Pais - QualityCourses	Jun 4, 2014	Jun 20, 2014	<input checked="" type="checkbox"/> running
<input type="checkbox"/> Ciudades - QualityCourses	Jun 4, 2014	Jun 20, 2014	<input checked="" type="checkbox"/> running
<input type="checkbox"/> Escuelas - QualityCourses	Jun 4, 2014	Jun 20, 2014	<input checked="" type="checkbox"/> running
<input type="checkbox"/> Reserva ahora - QualityCourses	Jun 4, 2014	Today	<input checked="" type="checkbox"/> running

Para ello se crea un *snapshot* para cada una de las paginas que queremos analizar, el programa funciona añadiendo un pequeño código script a las paginas para que comience a detectar las interacciones de los usuarios.

Datos obtenidos del *snapshot* de la Home page.

The screenshot shows the homepage of QualityCourses.com with several heatmaps indicating user click activity. The heatmaps are most prominent on the language selection buttons (English, Spanish, Italian, German, French, Portuguese), the 'LEARN', 'LOWEST PRICE', and 'DISCOVER' icons, and the four recommended language school cards. The footer contains navigation links and contact information.

Home - QualityCourses • <http://www.quality-courses.com/>
created 4-6-2014 • 16 days 15 hours 51 mins • 782 visits, 1102 clicks

Qualitycourses Language of Page: English Contact (+34) 933 189 153 / 902 022 790

Language courses abroad

What language would you like to learn?

- English
- Spanish
- Italian
- German
- French
- Portuguese

LEARN
Find the language course that suits you.

LOWEST PRICE
Minimum price guaranteed.

DISCOVER
Live an incredible experience abroad.

- Juniors Courses
- Work and Study
- Business Courses
- Long stay courses

Recommended language schools

School	2 weeks from	Book from
ILAC Toronto	419 €	104.75€
Embassy London	737 €	221.10€
Zoni Language Miami	415 €	124.50€
Campus Internacional Hispania	357 €	107.10€

© QualityCourses 2003-2014 - First Class Education S.L. - Via Laietana 57 pral 11 - 08003 Barcelona, España - CIF: B-66105107

Esta primera captura nos muestra las zonas donde los usuarios de la web han hecho click. Mostrando por colores o zonas de calor aquellas donde hay mas clicks

repetidos. Podemos observar que hay tres zonas bien diferenciadas donde se reciben el mayor numero de click. Una es el menú de selección de idioma de la pagina, y las otras dos son el idioma inglés y el idioma español.

Home - QualityCourses • <http://www.quality-courses.com/>
 created 4-6-2014 • 16 days 15 hours 51 mins • 782 visits, 1102 clicks

Qualitycourses Language of Page: English - Contact (+34) 933 189 153 / 902 022 790

Language courses abroad

What language would you like to learn?

English Spanish Italian German French Portuguese

LEARN Find the language course that suits you.

LOWEST PRICE Minimum price guaranteed.

DISCOVER Live an incredible experience abroad.

Juniors Courses Work and Study Business Courses Long stay courses

Recommended language schools

School	Location	Rating	Best value for money	Book from
ILAC Toronto	Toronto	★★★★★	Best value for money.	104.75€
Embassy London	London	★★★★★	Best value for money.	221.10€
Zoni Language Miami	Miami	★★★★★	Best value for money.	124.50€
Campus Internacional Hispania	Valladolid	★★★★★	Best value for money.	107.10€

About us Contact Terms & Conditions Minimum price guaranteed Blog

© QualityCourses 2003-2014 - First Class Education S.L. - Via Laietana 57 pral 1* - 08003 Barcelona, España. - CIF: B-66105107

En esta segunda captura se muestran las zonas de calor donde el usuario hace scroll y se queda más tiempo viendo en pantalla. Se puede observar que la zona donde aparecen los idiomas a estudiar es la zona donde más se detienen los usuarios, que es precisamente lo que se quiere conseguir.

También se puede ver que los usuarios hacen una pequeña parada en los otros productos que se ofrecen, y que difícilmente llegan a bajar del todo en la página.

Language courses abroad

What language would you like to learn?





LEARN
Find the language course that suits you.







LOWEST PRICE
Minimum price guaranteed.



DISCOVER
Live an incredible experience abroad.



Recommended language schools

 <p>ILAC Toronto Toronto ★★★★★ Best value for money. Book from: 104.75€ More details</p>	 <p>Embassy London London ★★★★★ Best value for money. Book from: 221.10€ More details</p>	 <p>Zoni Language Miami Miami ★★★★★ Best value for money. Book from: 124.50€ More details</p>	 <p>Campus Internacional Hispania Valladolid ★★★★★ Best value for money. Book from: 107.10€ More details</p>
--	---	--	--

Por último la tercera captura muestra las zonas donde se han realizado acciones, si estamos en la propia aplicación de *crazyegg* podremos ver el numero de clicks en cada zona y la procedencia de los usuarios que han hecho click allí.

Todas las capturas realizadas con *crazyegg* en las distintas secciones de las paginas están también incluida en los archivos anexos, en la carpeta de usabilidad.

Otra de las maneras que se han encontrado para medir usabilidad y también medir la repercusión de nuevos botones, o llamadas a la acción que se implementan es mediante Analytics de google, en el cual se pueden establecer eventos para averiguar cuánto porcentaje de usuarios hace click en algún punto.

De esta manera estableciendo eventos en cada una de las nuevas cosas que vamos añadiendo a la web somos capaces de ver si están teniendo el resultado que pensamos que iba a tener.

Categoría de evento	Total de eventos	% Total de eventos
1. NoBounce	11.106	92,67%
2. menu izq ciudad o pais	368	3,07%
3. infocourses	201	1,68%
4. Zopim Livechat	185	1,54%
5. vermapa	106	0,88%
6. contacto	16	0,13%
7. Ofertas	2	0,02%

Añadir eventos a la cuenta de analytics es sencillo, basta con añadir una línea de código a la zona que queremos medir dentro de una función onclick=" " y esta línea de código tendrá algunos parámetros como categoría del evento, nombre, o incluso si queremos darle un valor monetario a este evento.

Por ejemplo una de las aplicaciones que se hacen en la web es para medir cuanta gente hace uso del botón de mas información de los cursos en la página de ciudad y país con la línea de código siguiente:

```
onClick="_gaq.push(['_trackEvent', 'infocourses', 'masinfocourses', 'informacion de cursos ciudad o pais'])"
```

6. SEO.

6.1. Velocidad de carga.

6.2. Títulos.

6.3. Mejora de contenidos

6.4. Metas

6.5. Snippets

6.6. Blog

6. SEO

La mayor parte del tráfico web de la empresa viene por tráfico orgánico, con lo cual el SEO es de vital importancia para conseguir clientes en una empresa que trabaja online.

Uno de los mayores retos y objetivos de este proyecto es la recuperación del posicionamiento web y para ello se cuida detalladamente muchos elementos y recomendaciones para SEO.

Cuando se habla de SEO se puede dividir en dos partes, SEO onpage o SEO técnico y SEO offpage. Aquí hablaremos sobre el SEO onpage que es el que nos incumbe en la programación y el código a la hora de realizar la aplicación, pero hay que tener en cuenta otros factores como el SEO offpage que también influyen mucho en el posicionamiento, como por ejemplo los Enlaces Entrantes o Link Building y como sean este tipo de enlaces Entrantes ó "Backlinks"...etc.

Para todas las recomendaciones y comentarios nos basaremos en las pautas de google que es el principal buscador mundial, y para el cual se quiere posicionar la web.

6.1 Velocidad de carga

En primer lugar hablaremos de la velocidad de carga, un factor que aunque no se sabe en qué medida, influye en el posicionamiento web, una web que carga rápido y no hace esperar al usuario es bien considerada por google.

Con esta nueva web se ha conseguido mejorar considerablemente la velocidad de carga respecto de la anterior, para medirlo podemos utilizar la herramienta de google, PageSpeed Insights, donde se obtiene con la nueva web una puntuación de 73 / 100. También nos da algunas recomendaciones mediante las cuales podemos mejorar incluso esta puntuación.

La anterior web se encontraba por debajo de 50 puntos.



The image shows a screenshot of the Google PageSpeed Insights tool. On the left, there is a summary box with a score of 73 / 100 and the title 'Resumen de sugerencias'. Below this, there are two sections of recommendations:

- Elementos que debes corregir:** Eliminar el JavaScript que bloquea la visualización y el CSS del contenido de la mitad superior de la página. (Mostrar cómo corregirlo)
- Elementos que puedes plantearte corregir:** Minificar JavaScript. (Mostrar cómo corregirlo)

On the right side of the screenshot, there is a laptop displaying the website being analyzed, which is a language learning site with the heading 'Language courses abroad' and several circular icons representing different languages.

6.2 Títulos

Otro aspecto que se tiene muy en cuenta desde el momento del diseño de la interfaz es la estructuración del contenido, teniendo en todas las paginas un <h1> con las principales palabras clave para las que se quiere posicionar esa pagina, en segundo lugar un <h2> también con palabras clave pero menos importantes, y <h3> para los demás títulos.

Google tiene en cuenta las palabras clave incluidas en los títulos de las paginas, dando prioridad a los h1 y bajando a partir de ahí.

<h1>



<h2>

Estudiar Inglés en Inglaterra

<h3>



Nuestras escuelas de inglés en Inglaterra.

Todas las paginas siguen una estricta estructura en cuanto a títulos, utilizando en ellos las palabras clave que se quieren potenciar en esa página.

6.3 Mejora del contenido

Cada vez mas google tiene en cuenta el contenido de los sitios, y pide que sea contenido original y relevante para el usuario. Por ello en esta nueva aplicación se ha intentado mejorar el contenido de las paginas. Sobre todo las páginas de ciudad y país que son las que nos interesa posicionar y que peor contenido tenían.

Para ello se crean nuevos textos e información que es redactada y mostrada también de forma correcta, resaltando aquellas palabras clave que creamos convenientes con negritas y también incluyendo links dentro del texto hacia paginas de nuestra propia web, para facilitar al usuario la navegación.

Estudiar Inglés en Inglaterra

★★★★★ 4.7/5.0 (74)

Si decides realizar un **curso de inglés en Inglaterra**, **QualityCourses** te invita a hacerlo descubriendo por qué es uno de los países más populares del mundo...Por su **cultura**, por sus **tradiciones**, sus peculiaridades, sus contrastes, sus **paisajes**... Inglaterra no deja indiferente a nadie y es el destino favorito para aquellos que quieren **aprender inglés**.

Su capital, **Londres**, es una de las ciudades europeas más famosas y de las más visitadas del mundo, dónde se encuentran todos los iconos y tópicos del país, como los autobuses rojos de dos pisos, las **cabinas de teléfonos**, los guardias de la reina, el **Big Ben**, etc.

Aun así, **Inglaterra** está llena de ciudades para todos los gustos, así que hay **cursos de inglés en Inglaterra** para cada persona, porque cada ciudad tiene un encanto especial y con la tradición inglesa siempre presente.

En definitiva de lo que se trata es de generar contenido original y de calidad.

6.4 Metas

Uno de los puntos fuertes del SEO durante años son las metas, más concretamente la *meta title* y la *meta description*. Estas son unas etiquetas *html* que indican un titulo y una descripción de la pagina en concreto.

Estas metas están optimizadas también para cada una de las *url*, incluyendo palabras clave y descripciones del contenido, teniendo incluso en cuenta el numero máximo de caracteres que google recomienda para ellas.

```
57 <META NAME="Title" CONTENT="Cursos de inglés en Inglaterra - QualityCourses">
58 <META name="Description" CONTENT="QualityCourses te ofrece económicos cursos de Inglés con Precio Mínimo Garantizado en escuelas de idiomas de las principales ciudades de Inglaterra">
```

6.5 Snippets

Una de las novedades en los últimos tiempos en cuanto a la manera en que google muestra sus resultados en el buscador son los Snippets, que es el recuadro donde aparece cada web en los resultados de búsqueda, entre las novedades que se aprecian están la utilización de imágenes, o estrellas de valoración que aumentan el porcentaje de click en un resultado.

En este caso lo que se ha hecho en la web es preparar el contenido para que en las búsquedas de google aparezcan los resultados con las estrellas de valoración de los destinos y las escuelas.

Cursos de inglés en Londres - Quality-courses.com

www.quality-courses.com/cursos/ingles-inglaterra/londres/ ▾

★★★★★ Valoración: 4,9 - 33 votos

Cursos de **inglés en Londres**: Aprende idiomas con el Precio Mínimo Garantizado en los programas intensivos de **QualityCourses** en escuelas de idiomas en ...

Manuel Angel Rodríguez Izquierdo ha hecho +1 en esto

Google tarda algún tiempo en mostrar estos resultados una vez implementados en el código de la web, incluso es posible que no se muestren, pero de momento se han conseguido mostrar para ciudades y países.

6.6 Blog

Para conseguir tráfico es importante también que la empresa tenga un blog, y que este nutra de enlaces hacia la web principal y sus productos.

En este caso se cuenta con un blog wordpress instalado en el servidor de la web, donde las personas encargadas generan contenido periódicamente. Todo este contenido ha de ser original y de calidad para que google comience a dar votos positivos al dominio.

Con todos estos pequeños detalles y además el trabajo que se realice en la parte de SEO offpage se espera conseguir que con el tiempo la web vaya ganando posicionamiento. Hay que dejar claro que esto no ocurre de un día para otro, y que la competencia en este sentido es muy elevada, por ello también hace falta un plan para generar nuevos contenidos que puedan ser optimizados para según qué nicho de palabras clave donde la competencia no sea tan alta.

7. Analítica web .

7.1. Visitas y porcentaje de rebote.

7.2. Comercio electrónico.

7.3. Objetivos.

7.4. Adquisición.

7.5. Eventos.

7. Analítica web

Cada vez más el avance en las nuevas tecnologías permite medir, analizar y sacar conclusiones para luego tomar decisiones. Además todo lo que sea información y recogida de datos es valioso para las empresas.

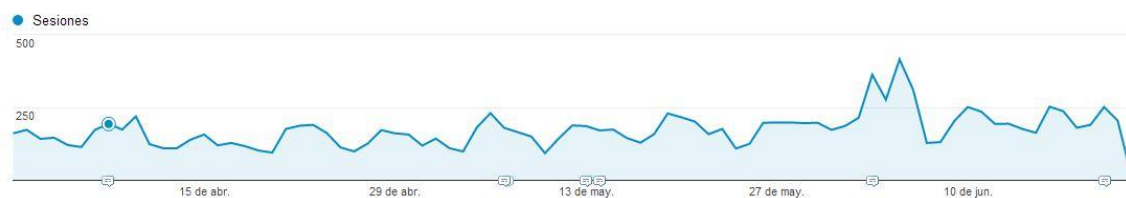
Por ello en este proyecto se ha querido realizar mediciones de todo lo que nos pueda ayudar a mejorar, a ver las pautas que sigue el usuario y a obtener datos que nos ayuden en las tomas de decisiones.

Estos datos son validos también para analizar si la nueva aplicación o los cambios aplicados en ella están funcionando como se espera.

Para la medición de estos datos se utiliza la herramienta Analytics de Google que mediante la implementación de un código en las páginas de la web que queremos analizar nos permite medir gran cantidad de datos.

7.1 Visitas y porcentaje de rebote

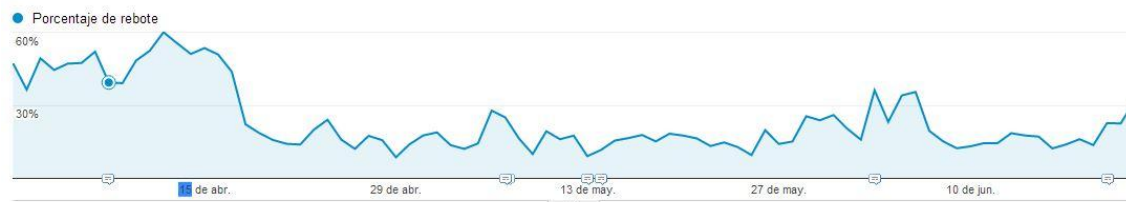
Comencemos por la fecha en que se publica al publico la nueva web, el 08/04/2014 y a partir de aquí se comienza a analizar comparando con las fechas anteriores y viendo la progresión. Todas las lecturas de las capturas siguientes son tomas desde el 01/04/2014 hasta el 21/06/2014.



Este es un grafico de las visitas a la web, podemos ver a simple vista un ligero aumento desde el día marcado, día 08/04/2014

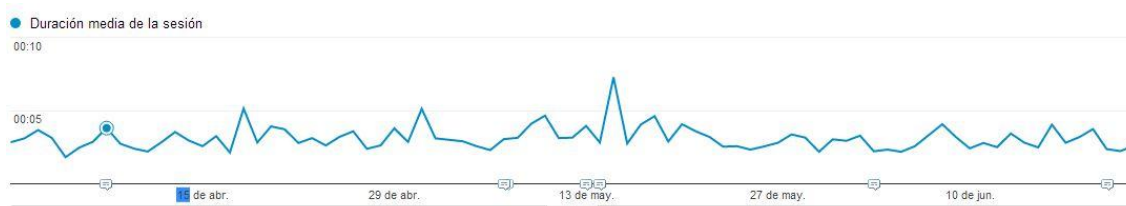
Esta herramienta nos permite marcar los días en los que se realiza algún cambio que puede ser importante y así poder medir los cambios a partir de esa modificación.

Uno de los datos más importantes que podemos obtener con analytics es el porcentaje de rebote de nuestra web. Esto nos dice el porcentaje de usuarios que se van de la web sin realizar ninguna acción, y era uno de los puntos clave que se querían mejorar con la nueva aplicación.



Como podemos observar en el grafico a los pocos días de tener la nueva aplicación online el porcentaje de rebote baja considerablemente , pasando de estar por encima del 50% a estar por debajo de un 25%.

El siguiente aspecto que debemos tener en cuenta es la duración media , el tiempo que los usuarios están en la aplicación.



En este caso en la actualidad se tiene una duración media de 03:00 minutos, se trata de un tiempo que se considera bueno, y no supone ninguna mejoría notable con respecto a la anterior web.

Pero donde si se ve una pequeña mejoría es en el número de páginas vistas por visita. Situándose en una media de 2,70 páginas por sesión.



Los datos recogidos nos permiten saber incluso el porcentaje de nuevos visitantes frente a visitantes que repiten y otros como idioma, país, navegador...etc



Datos demográficos		Idioma	Sesiones	% Sesiones
Idioma		1. es	3.970	27,87%
País/territorio		2. es-es	2.862	20,09%
Ciudad		3. en-us	1.904	13,36%
Sistema		4. it-it	961	6,75%
Navegador		5. it	850	5,97%
Sistema operativo		6. fr	721	5,06%
Proveedor de servicios		7. fr-fr	449	3,15%
Móvil		8. es-419	300	2,11%
Sistema operativo		9. de	274	1,92%
Proveedor de servicios		10. en	239	1,68%

[ver todo el informe](#)

También se pueden localizar las páginas de entrada y de salida de las visitas, y con ello localizar las paginas que pueden tener algún problema si tienen un porcentaje de rebote o salidas alto.

7.2 Comercio electrónico.

Una de las partes más importantes de la medición es la posibilidad de medir objetivos de comercio electrónico, lo cual no da el porcentaje de conversiones que tiene nuestra aplicación. Es decir el porcentaje de personas que compran a través de la web.

Estas mediciones están configuradas en el código de analytics para que cuando un cliente realice una compra esta sea registrada en el sistema, incluyendo el id de la venta y la cantidad en euros del pago realizado.



En este caso es imposible realizar comparaciones con fechas anteriores porque antes no estaba bien configurado el código de analytics para registrar las compras. Pero podemos ver que hay un porcentaje de conversiones del 0.20% , el cual se considera bajo, y es uno de los puntos que hay que intentar mejorar para mejorar ese porcentaje.

7.3 Objetivos

A parte del comercio electrónico, en las mediciones se pueden establecer también otros objetivos como por ejemplo el envío de formularios de contacto. Que quiere decir que se ha conseguido uno de los objetivos para los que se quiere la web, que es que los clientes entren en contacto.

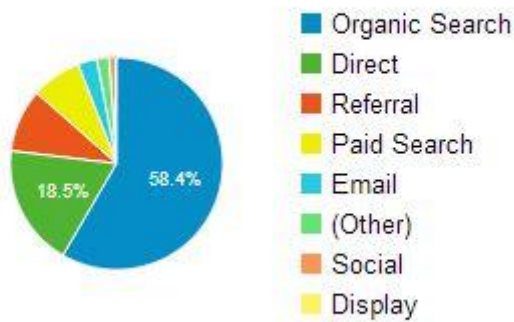
Los objetivos son configurados el 03/05/2014



7.4 Adquisición

Con estos datos se permite también saber cuál es la fuente del tráfico que llega a la web, y con ello poder medir también campañas de publicidad, redes sociales, blog...etc.

Canales principales



Viendo estos gráficos se pueden llegar a algunas conclusiones rápidas, como por ejemplo que hay que mejorar el tráfico procedente desde las redes sociales y seguir manteniendo el seo ya que el mayor porcentaje de tráfico sigue siendo de búsqueda orgánica.

7.5 Eventos

Como ya se hablo en el apartado de usabilidad, analytics es capaz de medir eventos que se establecen en la web, y mediante los cuales obtener datos de zonas que queremos analizar.

Categoría de evento	Total de eventos	% Total de eventos
1. NoBounce	20.710	93,07%
2. menu izq ciuda o pais	508	2,28%
3. Zopim Livechat	447	2,01%
4. infocourses	293	1,32%
5. Banner	116	0,52%
6. vermapa	114	0,51%
7. Videos	44	0,20%
8. contacto	16	0,07%
9. Ofertas	4	0,02%

[ver todo el informe](#)

En general la analítica web es una parte fundamental en las empresas online, y una de las herramientas más potentes que se pueden tener. Cuantos más datos se recogen mas control se tiene y mejores decisiones se pueden tomar. Cada día los datos son más valiosos para las empresas y saber interpretarlos nos puede ayudar a mejorar las aplicaciones y obtener mayores beneficios.

8. Conclusiones.

8.1. Técnico.

8.2. Madurez del código.

8.3. Diseño.

8.4. Otras áreas.

8.5. Puntos débiles.

8.6. Futuro de la aplicación.

8. Conclusiones

Los resultados obtenidos a la finalización de este proyecto son muy satisfactorios y consideramos que se han conseguido los objetivos que se marcaron al principio del mismo.

En primer lugar mostrar el agradecimiento a la empresa QualityCourses por la oportunidad que nos ha brindado al dejar en nuestras manos un proyecto tan interesante, en segundo lugar dar las gracias también a los trabajadores que durante este tiempo han llevado a cabo tareas externas a este proyecto pero que han sido utilizadas en él, como por ejemplo la redacción de contenidos, traducciones...etc

La gestión de un equipo de trabajo ha sido uno de los puntos en que más nos ha enriquecido este proyecto, llegando a tener a nuestra disposición hasta 3 personas en la empresa, que teníamos que coordinar para dar tareas de redacciones, traducciones o búsqueda de contenido. Y nos hemos topado con algunas dificultades propias del trabajo en equipo como puede ser descoordinación o desmotivación, pero que se han sabido suplir con efectividad.

8.1 Técnico

El apartado técnico es el punto fuerte de las conclusiones, ya que ha sido todo un reto el desarrollo de una aplicación de esta envergadura totalmente desde cero y en el cual se han adquirido muchos conocimientos, con este apartado estamos altamente satisfechos a pesar de que en ocasiones durante el desarrollo del proyecto se tuvieron problemas que gracias a la constancia y a la comunidad online se supieron solventar.

Es por ello que en un proyecto de estas características nos gustaría resaltar la importancia de saber buscar y apoyarse en las comunidades de programadores online que nos pueden brindar mucha ayuda cuando se está atascado en algún punto. Forma parte de ello también la elección de frameworks o lenguajes de programación que tengan a sus espaldas una buena comunidad de gente dispuesta a ayudar.

8.2 Madurez del código

El tiempo hace que el código de programación vaya madurando poco a poco y nos vamos dando cuenta de pequeños errores que vamos solventando o también partes del código que repetimos y podíamos haber integrado en una función.

En este apartado la aplicación necesita de un tiempo en ejecución para ir puliendo todo lo posible el código, y nos vamos dando cuenta de pequeñas cosas que podríamos hacer de manera más simple o más limpia, pero creemos que esto forma parte de la madurez de la aplicación a lo largo de su tiempo de vida.

8.3 Diseño

En el apartado de diseño, aunque este es un punto más personal, creemos que hemos conseguido un diseño atractivo, limpio, moderno y que da confianza al usuario final de cara a una posible compra. Lo que supone un gran cambio con respecto a la anterior versión de la web.

8.4 Otras áreas

Durante el tiempo de realización del proyecto hemos tenido la suerte de realizar algunas reuniones semanales o mensuales con expertos en distintas áreas y aprender de ellos lo máximo posible para aplicarlo en la web que realizamos.

Entre ellas tenemos que destacar especialmente todos los conocimientos obtenidos en SEO para la cual le dedicamos algunas reuniones semanales y de la que salimos con unos conocimientos altamente superiores a los que teníamos cuando comenzamos. También interesante todo lo aprendido en cuanto analítica web y lo más importante es lo conscientes que somos ahora de que estos dos apartados son muy importantes e interesantes cuando nos disponemos a realizar una aplicación web que queremos que triunfe.

Aunque ha habido algunos cambios positivos, nos es imposible mostrar unos resultados de mejoría en cuanto a SEO ya que ahora que está el trabajo hecho hay que esperar a que google vaya poco a poco recuperando posiciones si todo está bien implementado.

8.5 Puntos débiles

En todo lo que ha conllevado este proyecto creemos que uno de nuestros puntos débiles es el apartado de usabilidad, para el cual por falta de tiempo y recursos no se ha podido dedicar todo el tiempo que no hubiera gustado. Además de que las muestras que tenemos son sobre un número muy pequeño de usuarios.

Se ha hecho todo lo posible por aprovechar al máximo los pocos datos obtenidos que tenemos en este apartado, pero nos hubiese gustado disponer de más tiempo y recursos para la realización de más pruebas, a mayor número de usuarios y la obtención de mejores resultados.

8.6 Futuro de la aplicación

Cabe destacar que el proyecto no ha acabado aquí, puesto que se tiene preparado un plan para seguir mejorando la aplicación incluyendo en ella nuevas funciones y opciones que trataran de mejorar tanto la experiencia de usuario como la imagen de marca y las ventas de la empresa.

Entre algunas de estas mejoras que están planificadas se encuentran:

- Incluir nuevos servicios en el booking tool (transportes, actividades...etc).
- Sistema de códigos promocionales de descuento.
- Generación de presupuestos en pdf desde la web.
- Mejoras de las llamadas a la acción para incentivar la venta.
- Mejora del sistema de comentarios de los clientes.
- Incluir nuevos métodos de pago (paypal, financiación...etc.).

"La conclusión personal más importante es que termino este proyecto habiendo aprendido mucho y con más conocimientos y experiencia de los que tenía cuando comencé, y aunque aún falta mucho que aprender, creo que de eso se trata, de aprender de cada proyecto. "

9. Bibliografía.

10. Agradecimientos

9. Bibliografía

La bibliografía que se lista a continuación es una lista de las páginas consultadas para la documentación del proyecto. La bibliografía se ha estructurado en los temas tratados en esta memoria, para una fácil localización.

La competencia y referencias.

- <http://es.languagebookings.com/>
- <http://www.languagecourse.net/>
- <https://www.duolingo.com/>
- <http://web.archive.org/web/20130815101756/http://www.quality-courses.com/>

Diseño.

Versiones beta.

- www.aprenderidiomas.com
- www.abroadlanguages.com

Tipografía

- <https://www.google.com/fonts>

Color

- <https://kuler.adobe.com/es/FLAT-UI-color-theme-3247888/edit/?copy=true&base=2&rule=Custom&selected=3&name=Copy%20of%20FLAT%20UI&mode=rgb&rgbvalues=0.9254901960784314,0.9411764705882353,0.9450980392156862,0.9019607843137255,0.49411764705882355,0.1333333333333333,0.1803921568627451,0.8,0.44313725490196076,0.20392156862745098,0.596078431372549,0.8588235294117647,0.20392156862745098,0.28627450980392155,0.3686274509803922&swatchOrder=0,1,2,3,4>

Iconos

- <https://www.iconfinder.com/>

Maquetación

- <http://getbootstrap.com/>
- <http://eternicode.github.io/bootstrap-datepicker/?markup=input&format=&weekStart=&startDate=&endDate=&st>

artView=0&minViewMode=0&todayBtn=false&language=en&orientation=auto&multidate=&multidateSeparator=&keyboardNavigation=on&forceParse=on#sandbox

- <http://blog.bloque9.com/index.php/item/161-herramientas-utiles-para-bootstrap>
- <http://everythingfrontend.com/posts/star-rating-input-pure-css.html>

Programación backend

- <http://phptoolcase.com/>
- <http://www.forosdelweb.com/>
- <http://www.phpxtemplate.org/HomePage>
- <https://php.net/manual/es/index.php>
- <http://www.merinomartinez.com/2011/11/tarjetas-de-pruebas-para-tpv-virtuales/>
- http://vitaminaweb.com/ordenar-y-filtrar-resultados-de-consultas-mysql-con-php-sin-saturar-la-base-de-datos_969
- <http://www.funcion13.com/2012/04/25/cargando-contenido-ajax-con-jquery-usando-imagen-de-cargando/>

Usabilidad

- <https://www.crazyegg.com/>

SEO

- <http://www.marketingandweb.es/marketing/10-consejos-para-mejorar-el-seo-en-ecommerce/>
- <http://schema.org/>
- <https://support.google.com/webmasters/answer/189077?hl=es>
- <https://support.google.com/webmasters/answer/99170?hl=es>

Analítica web

- <http://www.google.es/intl/es/analytics/>
- <https://support.google.com/analytics/answer/1033068?hl=es>
- <https://support.google.com/analytics/answer/1032415?hl=es>
- <http://www.robertoballester.com/como-crear-un-evento-onclick-en-google-analytics/>

Url del proyecto

- <http://www.quality-courses.com>

10. Agradecimientos

En primer lugar quiero agradecer el apoyo de mi familia que ha sido la principal causante de que pueda estar aquí hoy. También a compañeros y amigos que me han ayudado en todo lo que han podido, con consejos, búsqueda de información, pruebas de usabilidad...etc.

Quiero agradecer también a la empresa QualityCourses por haberme dado la oportunidad de realizar este proyecto y haberme dejado libertad para poder plasmar en el todo lo que creía necesario. Y dentro de la empresa me gustaría agradecer primero a todo el equipo de trabajo que de forma indirecta a participado en que este proyecto haya visto la luz, y donde también me han enseñado todo lo que se ahora sobre ámbitos que antes no dominaba como SEO o Analítica web. En especial a Manuel Ángel Rodríguez Izquierdo, director de la empresa, que además han confiado en mí para formar parte del equipo.

Agradezco también el apoyo y las correcciones aportadas por mi directora de proyecto: Ana Gabriela Zuñiga Zarate, los cuales me han ayudado a mejorar el proyecto y sobre todo a la redacción de una forma correcta de esta memoria.

Para finalizar agradecer también al CITM la oportunidad de haber podido hacer un proyecto de mi libre elección, y a todos los profesores que he tenido durante los cuatro cursos que he pasado en el centro.