



Diseño y creación de una aplicación para crear listas ordenadas de centros educativos de Galicia

Memoria de Proyecto Final de Máster

Máster Universitario de Aplicaciones Multimedia

Autor: Marcos Chavarría Teijeiro

Consultor: Sergio Schvarstein Liuboschetz

Profesora: Laura Porta Simó

Junio 2018

Créditos



Esta obra está sujeta a una licencia de Reconocimiento-CompartirIgual
[3.0 España de Creative Commons](https://creativecommons.org/licenses/by-sa/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Diseño y creación de una aplicación para crear listas ordenadas de centros educativos de Galicia</i>
Nombre del autor:	<i>Marcos Chavarría Teijeiro</i>
Nombre del consultor/a:	<i>Sergio Schvarstein Liuboschetz</i>
Nombre del PRA:	<i>Laura Porta Simó</i>
Fecha de entrega (mm/aaaa):	06/2018
Titulación:	<i>Máster Universitario de Aplicaciones Multimedia</i>
Área del Trabajo Final:	<i>Trabajo de Fin de Máster</i>
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	Galicia, Concursos de asignación de centros, <i>Aplicación web.</i>
Resumen del Trabajo:	
<p>Anualmente los docentes gallegos deben participar en distintos concursos en donde eligen los centros en los que desean trabajar. En la actualidad se cubren dichos concursos a través de una aplicación introduciendo códigos de los centros educativos a mano. En este trabajo de fin de master se ha creado una aplicación que permita la creación de listas ordenadas de centros educativos gallegos con objetivo de facilitar la tarea anteriormente descrita a los profesores. Para ello diseñamos y implementamos una aplicación web usable tanto desde ordenadores de escritorio como desde móviles y tabletas y que permita filtrar y ordenar los centro siguiendo diversos criterios. El resultado es una aplicación que reúne en un solo sitio una gran cantidad de información de los centros educativos gallegos permitiendo a los docentes analizar la idoneidad de cada centro y construido con una arquitectura que permite su mejora y ampliación futura.</p>	
Abstract:	
<p>Galician teachers must every year participate to certain contest where they choose the education centers where they want to work. To apply to these contest, nowadays teachers should use a application where they type one by one every center code. Within this final master project an application which allows to create sorted lists of educational centers has been created. In order to achieve this we have designed and implemented a web application which can be used both from desktop computes as from mobile and tablets and which allows to filter and sort schools using different criteria. The result is a a application that gathers in one place a huge amount of information about Galician schools allowing teachers analyze if they like every center and built with an architecture which allows its future improvement.</p>	

Agradecimientos

Me gustaría aprovechar este espacio para agradecer la ayuda de aquellos que de una forma o otra han colaborado a que este proyecto se haga realidad.

Empezar dando gracias a quienes han proporcionado el soporte económico para poder realizar este máster, es decir, mis padres. Sin ellos sería imposible que hubiera llegado a donde estoy tanto como persona como profesional.

Agradecer además a todos los profesores no solo de este máster sino también de la ingeniería por haber dedicado tanto tiempo a que yo adquiriese los conocimientos de informática que espero haber podido demostrar en este trabajo.

Agradecer la colaboración de los compañeros y compañeras del grupo de Profesores de Matemáticas de Galicia por su constante aporte de ideas y de feedback y en especial a Alberto Fortes por su constante ayuda e implicación en el proyecto.

Por último y no por ello menos importante, agradecer a todos mis amigas, amigos y familiares todo su apoyo durante estos años tanto en los buenos como en los malos momentos.

A todos ellos, gracias!

Marcos Chavarría Teijeiro.

En Lugo a 3 de Junio de 2018.

Resumen

Anualmente los docentes gallegos deben participar en distintos concursos en donde eligen los centros en los que desean trabajar. En la actualidad se cubren dichos concursos a través de una aplicación introduciendo códigos de los centros educativos a mano.

En este trabajo de fin de máster se ha creado una aplicación que permita la creación de listas ordenadas de centros educativos gallegos con objetivo de facilitar la tarea anteriormente descrita a los profesores. Para ello diseñamos y implementamos una aplicación web usable tanto desde ordenadores de escritorio como desde móviles y tabletas y que permita filtrar y ordenar los centro siguiendo diversos criterios.

El resultado es una aplicación que reúne en un solo sitio una gran cantidad de información de los centros educativos gallegos permitiendo a los docentes analizar la idoneidad de cada centro y construido con una arquitectura que permite su mejora y ampliación futura.

Abstract

Galician teachers must every year participate to certain contest where they choose the education centers where they want to work. To apply to these contest, nowadays teachers should use a application where they type one by one every center code.

With this final master project an application which allows to create sorted lists of educational centers have been created. To do this we have designed and implemented a web application which can be used both from desktop computes as from mobile and tablets and which allows to filter and sort schools using different criteria.

The result is a a application that gathers in one place a huge amount of information about Galician schools allowing teachers analyze if they like every center and built with an architecture which allows its future improvement.

Lista de palabras clave

- Centros educativos.
- Concursos de asignación de centros.
- Aplicación web.

Índice

Capítulo 1: Introducción.....	11
1. Motivación.....	11
2. Pasos previos.....	12
3. Definición y objetivos generales del TFM.....	14
4. Estructura del resto del documento.....	14
Capítulo 2: Análisis y Metodología.....	16
1. Estado de arte de resolución del problema dado.....	16
2. Estado de arte de la tecnología a usar en el proyecto.....	19
3. Definición de objetivos del producto.....	20
4. Metodología.....	21
Capítulo 3: Planificación y Presupuesto.....	24
1. Planificación.....	24
2. Seguimiento de la planificación.....	26
2.1 Entrega 1 (PEC3).....	26
2.2 Entrega 2 (PEC4).....	27
2.3 Entrega final (PEC5).....	27
3. Presupuesto.....	28
Capítulo 4: Diseño.....	29
1. Arquitectura general de la aplicación.....	29
2. Diseño gráfico e interfaces.....	29
2.1 Inspiración.....	29
2.2 Prototipos Lo-Fi.....	34
2.3 Prototipos Hi-Fi.....	37
3. Tecnologías web usadas.....	38
Capítulo 5: Implementación.....	39
1. Implementación de la Base de Datos.....	39
2. Implementación del servicio OSRM.....	40
3. Implementación del front-end.....	41
3.1 Aprendizaje del framework Vue.js.....	41
3.2 Aspectos destacables del desarrollo.....	42

Capítulo 5: Implantación y pruebas.....	51
1. Instalación en el servidor.....	51
2. Optimización de velocidad de carga.....	52
3. Optimización SEO y promoción.....	53
4. Análisis de uso.....	54
Capítulo 6: Conclusiones y trabajo futuro.....	55
1. Conclusiones.....	55
2. Líneas de trabajo futuro.....	55
Bibliografía.....	57

Figuras y tablas

Índice de Figuras

Figura 1: Aplicación de la Xunta de Galicia para la participación en el concurso.....	10
Figura 2: Hoja de cálculo de centros de Galicia.....	11
Figura 3: Prototipo de la aplicación.....	12
Figura 4: Captura del análisis de Google Analytics para el prototipo realizado.....	12
Figura 5: Captura de imagen de la web de la Xunta de Galicia.....	15
Figura 6: Filtros de la web de la Junta de Castilla y León.....	16
Figura 7: Web del Ministerio de Educación.....	16
Figura 8: Diagrama de la arquitectura de la aplicación.....	27
Figura 9: Web Amazon (Escritorio).....	28
Figura 10: Web Idealista (Escritorio).....	28
Figura 11: Web Amazon (móvil).....	29
Figura 12: Diseño de baja fidelidad (Escritorio).....	29
Figura 13: Diseño de baja fidelidad (Móvil).....	30
Figura 14: Diseño de baja fidelidad de los centros.....	30
Figura 15: Diseño de panel cambiar de localización.....	31
Figura 16: Diseño panel exportar centros.....	31
Figura 17: Mockup (Escritorio).....	32
Figura 18: Mockup (Móvil).....	32
Figura 19: Captura de componente.....	37
Figura 20: Comparativa filtros en prototipo e en implementación.....	39
Figura 21: Captura de panel papelera.....	39
Figura 22: Captura de opciones de ordenar, reordenar y papelera.....	40
Figura 23: Captura de selector de página.....	40
Figura 24: Panel cambiar localización.....	41
Figura 25: Barra que advierte el cálculo del tiempo a cada centro.....	41
Figura 26: Resultados informes de carga.....	46
Figura 27: Resultados de la web en Google Analytics.....	47

Índice de Cuadros

Cuadro 1: Código de función del scraper.....	39
Cuadro 2: Ejemplo de centro en la base de datos.....	40
Cuadro 3: Servicio de Systemd para OSRM.....	41
Cuadro 4: Implementación de componente FilterList.....	43
Cuadro 5: Código función calcular tiempos.....	49
Cuadro 6: Ejemplo de uso de funciones de internacionalización.....	49
Cuadro 7: Ejemplo de cadenas traducidas.....	50

Cuadro 8: Configuración de VirtualHost NGINX.....	51
Cuadro 9: Activar GZip en NGINX.....	52
Cuadro 10: Tiempo de cache para imágenes y ficheros JavaScript.....	53

Capítulo 1: Introducción

En este capítulo realizaremos una primera aproximación al trabajo de fin de máster explicando por una parte la motivación del mismo y el origen del problema que se pretende responder, relatando a continuación los pasos dados para resolver el problema antes del inicio de este trabajo de fin de máster y terminando por detallar la estructura y contenido del resto de este documento.

1. Motivación

Antes de empezar a detallar los aspectos más técnicos de este trabajo de fin de master es conveniente saber de donde partimos y las motivaciones de este proyecto.

Me llamo Marcos Chavarría y trabajo de profesor de matemáticas en un instituto público de Galicia. He estudiado informática y tengo una cierta experiencia en la creación de páginas webs. Esto hace que en muchas ocasiones vea ciertos problemas que se pueden resolver usando las nuevas tecnologías. En este proyecto se pretende resolver o facilitar uno de estos problemas existentes.

Anualmente los docentes gallegos al igual que lo hacen en otras comunidades autónomas de España deben participar anualmente en una serie de concursos en los que se deciden en los centros educativos en los que trabajarán el próximo o los próximos cursos académicos.

En estos concursos los profesores y maestros indican los centros educativos en los que les interesa trabajar por orden de preferencia. La forma de indicar esta lista de centros consiste en introducir uno por uno cada uno de los códigos que referencia cada código en aplicaciones web muy rudimentarias. En la *Figura 1* se puede ver un ejemplo de este tipo de aplicaciones en las que se can introduciendo por orden cada uno de los códigos.

Para más inri estos códigos se obtienen a partir de documentos PDF que ofrece la *Consellería de Educación* como se puede ver por ejemplo en la convocatoria establecida en la ORDE do 13 de xuño de 2017 [1].

De esta forma nos enfrentamos a dos problemas: por una parte la dificultad de crear una lista de centros ordenada y por otra la dificultad de introducir los códigos en la web de la Consellería de Educación.

El objetivo principal de este trabajo de fin de máster es usar todos los conocimientos adquiridos durante el así como en la ingeniería informática que estudie previamente para construir una aplicación web que permita solucionar el primer problema. Es decir que permita ordenar y filtrar los centros educativos gallegos empleando varios criterios de forma que los profesores puedan ordenar los centros en dicha aplicación y después copiar los centros a la aplicación oficial de la Xunta de Galicia intentando si es posible integrar ambos servicios.

2. Pasos previos

Como se explicaba en el apartado anterior, el objetivo del proyecto es facilitar la elección de centros educativos a los profesores y maestros gallegos. No obstante, antes de llegar a este proyecto se dieron varios pasos que se detallarán a continuación.

En un primer momento, un grupo de profesores creó una hoja de cálculo que permitía ordenar los centros en función a la distancia en línea recta a unas coordenadas geográficas determinadas utilizando para ello la Fórmula de haversine [2]. Dicha hoja de cálculo se puede ver en la *Figura 2* y aunque ayudo en gran parte a conseguir un orden sobre las distancias, este orden no era exacto ya que el tiempo en llegar un determinado centro depende de la carretera y no solo de la distancia en línea recta.

	A	B	C	D	E	F	G	H	I	J
1										
2		http://www.coordenadas-gps.com/								
3	Latitude	43.3623436								
4	Lonxitude	-8.41								
5	Centros máis próximos									
6	Código centro	Nome centro	Enderezo	Concello	Provincia	Código postal	Teléfono	Latitude	Lonxitude	Cód locali
7	15027897	CIFP Paseo das Pontes	Rúa San Pedro de Mezonzo 4	Coruña (A)	Coruña (A)	15004	981160196	43.362698	-8.415876	15030
8	15005221	IES Plurilingüe Eusebio da Guarda	Praza de Pontevedra 20	Coruña (A)	A Coruña	15003	881960060	43.36831	-8.408111	15030
9	15005397	IES Fernando Wirtz Suárez	Rúa Cabaleiros 1	Coruña (A)	A Coruña	15006	881960260	43.35558	-8.40585	15030
10	15005211	IES Salvador de Madariaga	Rúa Paseo de Ronda 49	Coruña (A)	A Coruña	15011	881960235	43.370003	-8.420801	15030
11	15005233	IES Agra do Orzán	Rúa Alcalde Líaño Flores s/n	Coruña (A)	A Coruña	15010	881880080	43.362694	-8.425084	15030
12	15026421	IES Plurilingüe Rafael Dieste	Ronda de Outeiro 334	Coruña (A)	A Coruña	15011	981277911	43.3689	-8.424078	15030
13	15027770	IES A Sardiñeira	Avenida da Sardiñeira 41	Coruña (A)	A Coruña	15008	881960150	43.34991	-8.41622	15030
14	15005269	IES Urbano Lugrís	Avenida de Arteixo	Coruña (A)	A Coruña	15007	881960715	43.350506	-8.418542	15030
15	15025621	EPAPU Eduardo Pondal	Rúa da Educación 3 2º andar	Coruña (A)	A Coruña	15011	981278035	43.370552	-8.423923	15030
16	15005257	IES Ramón Menéndez Pidal	Rúa Instituto Zalaeta	Coruña (A)	A Coruña	15002	981203000	43.373623	-8.399839	15030
17	15005245	IES Monelos	Rúa Montes 62	Coruña (A)	A Coruña	15009	881880041	43.349415	-8.400689	15030
18	15005002	IES Ramón Otero Pedrayo	Avenida Monelos 10 A	Coruña (A)	A Coruña	15008	981286854	43.347878	-8.403661	15030
19	15005336	CEE Nosa Señora do Rosario	Rúa Ronda Camilo Jose Cela 2	Coruña (A)	A Coruña	15008	981283496	43.347874	-8.403603	15030
20	15005579	IES Plurilingüe Elviña	Rúa San José de Calasanz 6	Coruña (A)	A Coruña	15008	981242911	43.344296	-8.41085	15030

Figura 2: Hoja de cálculo de centros de Galicia

En un siguiente paso, otra persona creó una aplicación que partiendo de unas coordenadas geográficas, entregaba una lista de centros ordenada no solo por distancia en línea recta sino también por tiempo en coche desde un punto determinado. Para hacer esto se usó una API de una implementación de OSRM [3] que indicaba los tiempos en coche entre un punto geográfico y todos los centros educativos. Sin embargo, dicha aplicación solo permitía ordenarlos y no contemplaba todos los centros de Galicia, solo los de educación secundaria.

En este proyecto se pretende crear una aplicación que permita ordenar tanto por distancia como por tiempo en coche los centros educativos gallegos y además filtrarlos usando diversos parámetros como provincia, tipo de centro o estudios que en el se cursan.

Como prueba de viabilidad y realizada fuera del marco de este TFM se implementó un pequeño prototipo que se puede ver en la *Figura 3*. Este prototipo aunque incluye algunas de las características citadas, es poco vistoso y su usabilidad es muy baja por lo que no resolvería el problema que pretendemos resolver en su totalidad.



Figura 3: Prototipo de la aplicación

Pese a estas deficiencias el éxito del prototipo durante el último concurso fue notorio llegando a ser usada por más de 2000 usuarios únicos como se puede ver en la Figura 4.



Figura 4: Captura del análisis de Google Analytics para el prototipo realizado

3. Definición y objetivos generales del TFM

Una vez visto el problema que se quiere resolver y los pasos previos dados hasta el momento, llega el momento de realizar una primera definición de objetivos de este proyecto.

Partiendo de la prueba de viabilidad realizada necesitamos un sitio web que permita tanto filtrar como ordenar los centros educativos gallegos y que sea compatible con teléfonos móviles, tabletas y ordenadores de escritorio. Así mismo este producto deberá seguir un proceso de ingeniería que permita en el futuro su mejora y expansión en funcionalidades de forma simple.

Teniendo en cuenta esto definimos como objetivos para el producto creado durante este trabajo fin de máster los siguientes:

- Permitir ordenar centros por diversos criterios incluyendo distancia y tiempo en coche.
- Permitir filtrar centros usando diversos criterios incluyendo entre ellos el tipo de enseñanza que se imparten en cada centro.
- Permitir reordenar de forma manual los centros.
- Permitir eliminar los centros que no nos interesan.
- Compatible y usable en ordenadores de escritorio, tabletas y teléfonos móviles.
- Arquitectura que permita una expansión en funcionalidades futura.
- Diseño atractivo.

Por otro lado marcamos como objetivos personales del autor de este proyecto:

- Aprender el *framework* JavaScript para aplicaciones *front-end*.
- Profundizar en el aprendizaje de SCSS así como de algún *framework* relacionado para el desarrollo de aplicaciones *front-end*.

4. Estructura del resto del documento

Como se destacaba anteriormente uno de los objetivos de este trabajo final de máster es que el producto creado sea fruto de un proceso de ingeniería de software. Es por ello por lo que en el resto de capítulos detallaremos los pasos dados en dicho proceso.

En el capítulo dos se tratarán aspectos relativos a la fase de **análisis**. Veremos el estado del arte tanto en cuanto al producto que queremos desarrollar como a la tecnologías que usaremos para ello. Terminaremos con una definición formal de objetivos y del alcance del proyecto y con una explicación de la metodología que se seguirá durante el mismo.

En el capítulo 3 se explican los aspectos relativos a la **planificación**. Se detallará la división en tareas del proyecto a realizar y la metodología de desarrollo empleada. Por último se realizará un

presupuesto de lo que costaría este proyecto en un entorno real.

En el siguiente capítulo trataremos la fase de **diseño** de la aplicación. Por una parte explicaremos la estructura general de la aplicación y por otra el diseño de interfaces.

En el capítulo 5 se explicarán detalles técnicos del proceso de **implementación** de los diseños realizados durante la fase anterior. Se especificarán dichos detalles para cada parte del producto.

El capítulo 6 se dedicará a mostrar el proceso de **implantación** de la aplicación desarrollada así como las diferentes optimizaciones realizadas.

El capítulo 7 se realizarán unas **conclusiones** del proyecto además de tratar vías de trabajo futuras y posibles mejoras al producto realizado.

Por último se incluyen las **referencias bibliográficas** empleadas durante este documento.

Capítulo 2: Análisis y Metodología

En este capítulo se relatarán todos los procesos realizados relacionados con la análisis. Se empezará detallando los resultados del proceso de investigación llevado a cabo en cuanto al estado del arte tanto en cuanto a la resolución del problema dado como a las tecnologías a usar para la resolución de este problema. A continuación se detallaran la especificación formal de los objetivos de este proyecto y se terminará explicando la metodología a seguir.

1. Estado de arte de resolución del problema dado

El propósito inicial de la aplicación es, de forma resumida, la creación de una plataforma que permita buscar y ordenar los centros educativos públicos de la Comunidad Autónoma de Galicia. Por ello buscaremos que aplicaciones existentes hay para este cometido no solo en Galicia sino también en otras comunidades.

La *Consellería de Educación* ofrece en Galicia una aplicación [4] para buscar centros educativos tanto públicos como privados y concertados pudiendo filtrarlos por tipo de centro, provincia y ayuntamiento así como el tipo de titularidad del centro. Se pueden exportar dichos datos en varios formatos.

Filtros para a busca
Indique polo menos un filtro e prema o botón Buscar

Centro
Código do centro
Tipo de centro: IES - Instituto de Educación Secundaria
Nome do centro
Provincia: Lugo
Concello: -- Todos --
Titularidade e concerto: Titularidade: Pública, Dependencia: -- Todos --
Enseño: Plan: -- Todos --, Réxime: -- Todos --, Enseño: -- Todos --

Localización
Mapa Satellite

37 elementos atopados, amosando dende o 1 ata o 10.
[Primeiro/Anterior] 1,2,3,4 [Seguinte/Ultimo]

Código	Nome	Enderezo	Concello	Provincia	Cód. postal	Teléfono	Ver detalle
27007247	IES Francisco Daviña Rey	Rúa Doutor Casares 63-67	Monforte de Lemos	Lugo	27400	982828200	
27015761	IES Ánxel Fole	Rúa Angelo Colocci s/n	Lugo	Lugo	27003	982870991	
27007259	IES Río Cabe	Rúa San Pedro s/n	Monforte de Lemos	Lugo	27400	982828034	
27015311	IES A Pinguela	Estrada de Sober s/n	Monforte de Lemos	Lugo	27400	982828051	
27012048	IES Xograr Afonso Gómez de Sarria	Rúa Ánxel Fole, s/n (Zona escolar)	Sarria	Lugo	27600	982870801	
27006553	IES Xoán Montes	Rúa Montevideo	Lugo	Lugo	27001	982870407	
27016704	IES Leiras Pulpeiro	Rúa Orquídea 45	Lugo	Lugo	27004	982870419	

Figura 5: Captura de imagen de la web de la Xunta de Galicia

De cada centro se muestra el código, nombre, dirección y teléfono. La opción de exportar nos muestra además las coordenadas geográficas del centro. En la *Figura 5* podemos ver que la web es muy poco vistosa y es muy poco apta para móviles. Además la web no permite ordenar los centros siguiendo algún criterio como el de distancia en línea recta.

DATOS DEL CENTRO

Nombre:

Código:

Provincia: SALAMANCA ▼

Localidad: Seleccione una localidad... ▼

Titularidad: Seleccione la titularidad ... ▼

ENSEÑANZAS

Nivel Educativo: Educación Secundaria Obligatoria (ESO) ▼

Familia: Seleccione una familia... ▼

Estudios: Seleccione unos estudios... ▼

Régimen matriculación: Seleccione el régimen de matriculación... ▼

Texto libre:

SERVICIOS A LAS FAMILIAS

Transporte
 Comedor
 Madrugadores
 Tardes en el cole
 Residencia/ Escuela hogar

OTROS SERVICIOS

Centro Bilingüe
 Jornada Continua
 Infoeduc@

Buscar
Limpiar

Figura 6: Filtros de la web de la Junta de Castilla y León

Por otra parte la Consejería de Educación de Castilla y León tiene una web similar [5] pero que incluye la posibilidad de filtrar os centros por determinados servicios que se ofrecen a las familias como pueden ser Transporte, Comedor, escuela hogar o otros (Figura 6). También permite exportar los resultados y muestra campos similares a los de la web gallega.

Tanto la Consejería de Educación de Asturias [6], como la de Cantabria [7] como la del País Vasco ofrecen aplicaciones con el mismo propósito pero sin ninguna funcionalidad a destacar. De la misma forma el resto de comunidades autónomas ofrecen aplicaciones similares que permiten de

Programa de consultas

Consulta relativa a:

- Comunidad autónoma: **COMUNIDAD DE MADRID**
- Naturaleza: **Centro público**
- Tipo de Centro: **Todos**
- Denominación Genérica: **Todos**
- Subdivisión: **Todos**
- Localidad: **Todos**
- Nivel: **Bachillerato**
- Familia: **Todos**
- Enseñanza: **Todos**
- Centros seleccionados: **288**

Atrás
Inicio
Generar documento PDF

Provincia	Localidad	Denominación Genérica	Denominación Especifica	Código	Naturaleza
Madrid	Alcala de Henares	Instituto de Educación Secundaria	ALBÉNIZ	28042590	Centro público
Madrid	Alcala de Henares	Instituto de Educación Secundaria	ALKALA-NAHAR	28039682	Centro público
Madrid	Alcala de Henares	Instituto de Educación Secundaria	ALONSO DE AVELLANEDA	28000467	Centro público
Madrid	Alcala de Henares	Instituto de Educación Secundaria	ALONSO QUIJANO	28035305	Centro público
Madrid	Alcala de Henares	Instituto de Educación Secundaria	ANTONIO MACHADO	28000522	Centro público
Madrid	Alcala de Henares	Instituto de Educación Secundaria	ARQUITECTO PEDRO GUMIEL	28028301	Centro público

Figura 7: Web del Ministerio de Educación

cierta forma recuperar los centros educativos de cada una de ellas.

A nivel del Ministerio de Educación [8] existe una aplicación genérica que permite filtrar por el tipo de titularidad del centro, el tipo de enseñanza y de centro y la localización geográfica. Como podemos ver en la Figura 7, en los resultados se muestra la provincia, localidad, tipo de IES, un código identificativo y su titularidad.

Vemos entonces que aunque todas las comunidades ofrecen más o menos una plataforma para la búsqueda de centros educativos estas son en ocasiones muy precarias y anticuadas. Además están orientadas a la perspectiva del padre/madre que busca un centro para su hijo o hija o del alumnado que quiere matricularse en algún centro. De esta forma la perspectiva y preocupaciones del profesorado como ordenar los centros por distancia en coche no están cubiertas.

Además los diseños tienden a ser muy poco orientados al uso en tabletas y teléfono móviles por lo que este aspecto tampoco está cubierto.

2. Estado de arte de la tecnología a usar en el proyecto

En cuanto a la creación de páginas web, la web sigue una arquitectura cliente-servidor de forma que existe un servidor que provee una serie de documentos web que nuestro navegador interpreta y nos muestra por pantalla. Es por ello que en el renderizado de la aplicación intervienen estos dos elementos, cliente y servidor; y por lo tanto es conveniente hablar de tecnologías que se ejecutan en cada uno de estos elementos.

En cuanto a tecnologías del lado del servidor podemos distinguir aquellas destinadas al almacenamiento de datos es decir las **bases de datos** y aquellas destinadas a procesar dichos datos que sería el **back-end** de la aplicación. Por otro lado, en el lado del cliente tenemos las tecnologías del **front-end** de la aplicación que se encarga de como mostrar los datos.

Existen hoy en día incontables sistemas de bases de datos que podemos de forma genérica distinguir entre bases de datos **SQL** y bases de datos **No (solo) SQL**. Las bases de datos SQL son las que más tiempo llevan en funcionamiento y dan soporte a la mayor parte de casos de uso de forma eficiente. Podemos citar como ejemplos de bases de datos SQL, MariaDB y Oracle siendo estas dos una de las más usadas. Por otro lado, bajo la etiqueta de No-SQL se engloban muchos tipos diferentes de base de datos que han surgido en los últimos años y que son más eficientes en determinados casos de uso. Algunos de los ejemplos más conocidos de este tipo de bases de datos son CouchDB, MongoDB, Casandra o Redis.

En cuanto a tecnologías de back-end, existen numerosas soluciones dependiendo del lenguaje de programación. Si trabajamos en PHP los framework's mas conocidos son Laravel o Symfony o para proyectos escritos en Python podemos usar Django o Flask. No obstante, una de las opciones más comunes a día de hoy es usar como lenguaje del lado del servidor el mismo lenguaje que

usaremos del lado del cliente, es decir, usar JavaScript a través de Node.js. En cuanto a tecnologías que usan Node.js podemos hablar de Express o de Next.

Por último en cuanto a tecnologías front-end, estas son las que se encargan de mostrar los datos en el navegador y por lo tanto al final siempre se estarán usando HTML, CSS y JavaScript ya que estos son los lenguajes que entiende el navegador¹. Es por ello que lo más simple es siempre usar estos tres lenguajes para mostrar los datos que se procesan del back-end. Esta opción sin embargo es realmente tediosa y tiende a generar código de baja calidad por su complejidad. Por ello existen diversos frameworks tanto para CSS como para JavaScript para facilitar este proceso. En cuanto a frameworks CSS podemos citar entre otros:

- **Bootstrap** [10]: Creado por Twitter, es sin dudas el framework CSS más usado. Provee un sistema de grid y gran cantidad de componentes además es responsive y tiene muy buena documentación.
- **Foundation** [11]: Es una familia de frameworks creado por la empresa ZURB. Que cuenta con versiones tanto para webs como para e-mails. También tiene un sistema de grid y gran cantidad de componentes. Su documentación es muy buena aportando gran cantidad de recursos y ejemplos.
- **Semantic UI** [12]: Creado en 2013 por un desarrollador independiente, provee lo aportado por los anteriores centrándose en crear un código HTML conciso e intuitivo.

Unido a estos frameworks existen pre-procesadores de CSS como LESS o SASS que añaden nuevas funcionalidades al lenguaje permitiendo un mejor mantenimiento. Por otro lado, en cuanto a frameworks JavaScript podemos citar entre otros:

- **React** [13]: Creado por Facebook y liberado en 2013 se define como una biblioteca JavaScript para crear interfaces.
- **Angular** [14]: Creado por Google y liberado en 2010 se define como el framework MVW² superheróico de JavaScript.
- **VueJS** [15]: Creado por un ex-trabajador de Google en 2014 y ha tenido un gran crecimiento en los últimos años. Se define como un framework MVVM intuitivo, rápido y componible.

Las tres tecnologías que se citaron están basados en componentes y difieren fundamentalmente en el tipo de plantillas que utiliza para visualizar los datos, la versión de JavaScript que usan y a como funcionan internamente.

3. Definición de objetivos del producto

Durante esta sección estableceremos cuales son los objetivos del presente proyecto y el alcance

1 Actualmente se está empezando a usar el lenguaje *WebAssembly* [9] aunque su uso es más limitado en la actualidad.

2 Model-View-Whatever (<https://plus.google.com/+AngularJS/posts/aZNVhj355G2>)

que le pretendemos dar. De esta forma teniendo en cuenta lo visto en la sección anterior podemos definir los objetivos de este Trabajo de Fin de Máster como los siguientes:

- Listado de centros educativos de titularidad pública de Galicia.
- Mostrar información relevante de cada centro incluyendo algunos datos que faciliten la decisión del profesorado sobre que centros escoger.
- Permitir filtrar dichos centros por diversos criterios:
 - Provincia.
 - Tipo de Centro.
 - Estudios.
- Permitir ordenar os centros por diversos criterios.
 - Nombre.
 - Distancia en coche.
 - Tiempo en coche.
 - Manual.
- Diseño atractivo.
- Diseño con buena usabilidad.
- Diseño *responsive* para facilitar su uso en teléfonos móviles y tabletas (implementación con framework CSS).
- Implementación que facilite la extensibilidad de la aplicación (implementación usando un framework JavaScript).
- Implementación de la aplicación internacionalizada y accesible.

En cuanto al **alcance** de la aplicación aunque queda bastante definido por los objetivos anteriores se puede declarar como:

Creación de una aplicación web que funcione de forma correcta y sea usable tanto en ordenadores de escritorio como en móviles y tabletas y que tenga como funcionalidad la de listar centros educativos de Galicia permitiendo filtrarlos por provincia, tipo de centro y tipo de estudios y ordenarlos por nombre distancia y tiempo en coche desde un punto determinado así como modificar dicho orden de forma manual.

4. Metodología

En esta sección se describe la metodología llevada a cabo para la elaboración de este proyecto.

Tanto la estrategia seguida para abordar el problema como la metodología de desarrollo.

Gran parte del **trabajo** realizado para la construcción de esta aplicación estaba iniciado en mayor o menor medida para el prototipo realizado con anterioridad. Por ello las posibles aproximaciones eran, coger dicho prototipo y mejorarlo o realizar un producto de cero.

Se ha optado por una aproximación intermedia pues crearemos la página web desde cero pero aprovecharemos el trabajo hecho en ciertos ámbitos como en la interacción con el servicio de cálculo de rutas. Creemos que esta es la opción idónea pues se usa el trabajo y el conocimiento adquirido hasta ahora sin que la aplicación a crear contenga los vicios de un prototipo hecho de forma rápida.

En cuanto a la **metodología de investigación** se optó por mantener conversaciones informales tanto con compañeros de trabajo del centro como con otros profesores a través de un grupo de Facebook que agrupa a todos los profesores de enseñanza secundaria de Galicia.

En cuanto a la **metodología de desarrollo**, se entiende por metodología un conjunto de métodos o prácticas empleadas para la realización de una tarea, en este caso el análisis, diseño e implementación de la aplicación web.

La mayor parte de métodos usados en el día a día del desarrollo de software se refieren a un trabajo en equipo ya que esta es una disciplina que tiene una complejidad suficiente como para necesitar de varias personas para realizar un producto. Además generalmente se trata de productos que requieren un importante número de horas para ser implementados.

No obstante el caso a tratar no reúne muchas de las características anteriormente detalladas. Estamos hablando de un equipo de una persona que hace todo el trabajo o dos en el mejor de los casos incluyendo al mentor. Y la necesidad horaria del proyecto no es elevada: pensar que según la planificación anterior el proceso de análisis, diseño e implementación llevará menos de dos meses. Es por ello que decir que se usa en su totalidad una metodología como *Scrum* [16] o cualquier tipo de metodología *Agile* [17] que sin duda usaríamos en proyectos de otra magnitud sería en cierto modo no contar la realidad del proyecto.

Pese a lo anteriormente descrito, si que empleamos algunos elementos de metodologías como Scrum o eXtreme Programming [18]. Podemos destacar como elementos característicos de la metodología empleada los siguientes

De **eXtreme Programming (XP)** se han escogido las siguientes características:

- **Pertenencia Colectiva:** El programa que se está implementando es Software Libre y por lo tanto su código está disponible para todo aquel o aquella que quiera mejorarlo o adaptarlo a sus necesidades. En concreto, el código fuente del programa está alojado en la plataforma GitHub a través de la cual se pueden reportar fallos o enviar cambios que serán integrados en el programa.
- **Estándares de Programación:** En la elaboración del programa se intentó seguir un estilo de código fijo en todos los ficheros. De esta forma facilitamos a los nuevos desarrolladores la

lectura del código de nuestro programa.

- **Diseño Simple:** Intentamos solucionar los problemas empleando un diseño simple teniendo en ocasiones que refactorizar ciertas partes del código para integrar nuevas características.
- **Ciente en el sitio:** Desde que se llegó a un estadio del desarrollo suficientemente importante se subió el sitio web implementado a un servidor para que los clientes, es decir, los profesores y maestros gallegos; pudiesen probarlo e indicar correcciones o mejoras del mismo.

De **Scrum** se han escogido las siguientes características:

- **Entregas parciales:** Los *sprints* que se definen en Scrum tienen el mismo propósito que las pruebas de evaluación continuada (PEC) que la UOC exige en la materia del trabajo de fin de máster. Este propósito es sin duda dividir el trabajo total en varias entregas para simplificar su realización. La principal diferencia es que mientras con Scrum se define cada *sprint* en una reunión al principio del mismo, en la metodología empleada definimos todas las tareas al principio.
- **Lista de tareas:** Tenemos una lista de tareas a realizar con tiempos asignados que vamos realizando a lo largo de cada entrega parcial.
- **Demostraciones con los clientes:** Relacionado con la característica de XP, de cliente en el sitio, se realizan demostraciones periódicas con profesores conocidos para saber su opinión sobre la aplicación que se está implementando.

Capítulo 3: Planificación y Presupuesto

Una vez especificada las características del proyecto a desarrollar como se hizo en el capítulo anterior pasamos a realizar una planificación de la realización del proyecto. Por último se establecerá un presupuesto del proyecto en base a la planificación realizada.

1. Planificación

Durante sección se tratará, como su nombre indica aspectos relativos a la planificación del proyecto. Teniendo en cuenta los objetivos y el alcance propuestos en el capítulo anterior dividiremos el trabajo en actividades dándoles una duración y unas fechas de realización.

En la siguiente tabla se muestra el desglose de las actividades así como las sucesivas entregas que hay que realizar durante la realización del Trabajo de Fin de Máster. Se ha incluido también las tareas previas a la presenta entrega por considerarlas partes del proyecto aunque no se realicen en el marco del trabajo.

Para cada actividad se ha especificado su nombre, duración, fecha de inicio y fecha de fin. Hay que tener en que para la realización de los cálculos de duración de cada actividad se ha contabilizado **cada día como 2 horas** de trabajo debido a que actualmente estoy trabajando y es el tiempo diario del que puedo disponer de forma realista.

Nombre	Dur.	Inicio	Final
0. TRABAJO PREVIO	??	??	26/03/2018
<u>0.1 Análisis.</u>	27 días	??	??
0.1.1 Investigación sobre el estado del arte.	2 días	??	??
0.1.2 Recolección de información.	3 días	??	??
0.1.3 Realización del prototipo.	20 días	??	??
0.1.4 Puesta en marcha del prototipo y publicación del mismo a la comunidad educativa durante el último concurso.	1 día	??	??
0.1.5 Análisis de resultados del prototipo.	1 día	??	??
<u>0.2 Planificación.</u>	3 días	24/03/2018	26/03/2018
1. PEC3 (1º entrega)	28 días	27/03/2018	23/04/2018
<u>1.1. Diseño de la aplicación.</u>	7 días	27/03/2018	01/04/2018
<i>1.1.1. Búsqueda de información.</i>	2 días	27/03/2018	28/03/2018
<i>1.1.2. Creación de mockups a mano.</i>	5 días	29/03/2018	02/04/2018

1.1.1.1 Mockup de estructura general de la aplicación.	1 día	29/03/2018	29/03/2018
1.1.1.2 Mockup de modulo centro.	2 días	30/03/2018	31/03/2018
1.1.1.3 Mockup de barra filtros.	0.5 días	01/04/2018	01/04/2018
1.1.1.4 Mockup de barra búsqueda.	0.5 días	01/04/2018	01/04/2018
1.1.1.5 Mockup de panel exportar.	0.5 días	02/04/2018	02/04/2018
1.1.1.6 Mockup de panel cambio de localización.	0.5 días	02/04/2018	02/04/2018
1.1.2. Creación de prototipos de alta fidelidad.	15 días	03/04/2018	16/04/2018
1.1.2.1 Instalación de entorno de desarrollo.	1 día	03/04/2018	03/04/2018
1.1.2.2 Creación de estructura web básica.	1 día	04/04/2018	04/04/2018
1.1.2.3 Implementación de diseño usando framework CSS.	12 días	05/04/2018	16/04/2018
<u>1.2 Creación de la base de datos de centros</u>	7 días	17/04/2018	23/04/2018
1.2.1 Exportar base de datos Xunta de Galicia.	1 día	17/04/2018	17/04/2018
1.2.2. Creación de encuesta para enviar a centros.	2 días	18/04/2018	19/04/2018
1.2.3. Envío de encuesta a centros.	1 día	20/04/2018	20/04/2018
1.2.3. Recolección de resultados.	2 días	21/04/2018	22/04/2018
1.2.4. Exportar base e datos en formato JSON.	1 día	23/04/2018	23/04/2018
2. PEC4 (2º entrega)	28 días	24/04/2018	21/05/2018
<u>2.1 Aprendizaje conceptos básicos framework VUE.js</u>	4 días	24/04/2018	27/04/2018
<u>2.2 Implementación de la aplicación</u>	24 días	28/04/2018	21/05/2018
2.2.1 Configuración del entorno de desarrollo (IDE, Git, etc.)	1 días	28/04/2018	28/04/2018
2.2.2 Creación de estructura estructura básica.	1 día	29/04/2018	29/04/2018
2.2.3 Implementación del módulo centro.	4 días	30/04/2018	03/05/2018
2.2.4 Implementación de opciones de filtrado por provincias.	1 día	04/05/2018	04/05/2018
2.2.5 Implementación de opciones de filtrado por tipo de centro.	1 día	05/05/2018	05/05/2018
2.2.6 Implementación de opciones de filtrado por estudios.	1 día	06/05/2018	05/06/2018

2.2.7 Implementación de ordenado por nombre.	1 día	07/05/2018	07/05/2018
2.2.8 Implementación de reordenado a mano.	2 días	08/05/2018	09/05/2018
<i>2.2.9 Implementación de ordenado por distancia y tiempo.</i>	5 días	10/05/2018	14/05/2018
2.2.9.1 Configuración y puesta en marcha de servicio ORSM usando Docker y un servidor propio.	2 día	10/05/2018	11/05/2018
2.2.9.2 Implementación de funciones de cálculo de distancias.	2 días	12/05/2018	13/05/2018
2.2.9.3 Implementación de controles.	1 días	14/05/2018	14/05/2018
2.2.10 Implementación de panel cambiar localización.	2 días	15/05/2018	16/05/2018
2.2.11 Implementación de panel exportar lista de códigos.	1 día	17/05/2018	17/05/2018
2.2.12 Internacionalización de la aplicación.	1 día	18/05/2018	18/05/2018
2.2.13 Localización de la aplicación a Gallego y Castellano.	1 día	19/05/2018	19/05/2018
2.2.14 Revisión de estándares de accesibilidad.	2 días	20/05/2018	21/05/2018
3. Entrega Final	21 días	22/05/2018	11/06/2018
<u>3.1 Realización de pruebas con usuarios finales.</u>	2 días	22/05/2018	23/05/2018
<u>3.2 Corrección de incidencias.</u>	3 días	24/05/2018	26/05/2018
<u>3.3 Redacción y corrección de la memoria.</u>	13 días	27/05/2018	08/06/2018
<u>3.4 Creación de presentación.</u>	3 días	09/06/2018	11/06/2018

Esta es la planificación se realizó al principio del desarrollo del proyecto y por lo tanto está sujeta a innumerables cambios debido a la mala estimación de los tiempos que lleva cada tarea. En la siguiente sección comentaremos algunos cambios surgidos a lo largo del desarrollo del proyecto.

2. Seguimiento de la planificación

Como se comentaba al final del capítulo anterior el desglose de tareas con sus tiempos asignados es una estimación que a lo largo de la tarea sufrió diversas modificaciones. En esta sección se detallarán dichas modificaciones:

2.1 Entrega 1 (PEC3)

Durante la primera entrega el proceso de realización del prototipo de alta fidelidad llevó más tiempo del planificado.

Por otro lado al analizar en profundidad la base de datos proporcionada por la Xunta de Galicia se descubrió que esta aportaba la mayor parte de datos que se necesitaban y que sería más complicado obtener a través de una encuesta por necesitar esta última de la disposición de los centros educativos a rellenarla. Debido a esto en vez de crear una encuesta y enviarla a los centros se decidió crear una herramienta que analizara la base de datos de la Consellería extrayendo todos los datos de interés. Este nuevo planteamiento lleva más tiempo.

Los dos factores unidos causaron que no fuésemos capaces de entregar la base de datos completa al final de la primera entrega como estaba planificado y lo tuvimos que posponer a la segunda.

2.2 Entrega 2 (PEC4)

Unido al retraso acumulado de la anterior entrega se detectó que el proceso de aprendizaje del framework escogido para el desarrollo de la aplicación llevaba más tiempo del planificado.

Debido a esto y a que no queríamos volver a posponer tareas de una entrega a otra se decidió aumentar la carga de trabajo durante la fase final de la práctica y pasar de 2 horas diarias a 4 horas diarias hablando siempre en términos medios.

Con esta modificación conseguimos entregar todo el trabajo a tiempo.

2.3 Entrega final (PEC5)

Esta entrega se dedicó en su práctica totalidad a la redacción de esta memoria así como la creación de las presentaciones que requirieron más tiempo del planificado. En cuanto a la corrección de incidencias se dedicó la mayor parte del tiempo a la realización de optimizaciones a nivel SEO y de velocidad de carga así como la instalación de algunas herramientas de análisis.

Para poder completar la tarea con éxito al igual que sucedió en la entrega anterior, se tuvo que aumentar la carga de trabajo a 3 horas diarias (en términos medios).

3. Presupuesto

Una vez realizado la planificación de este proyecto se muestra un presupuesto estimado de los costes de la realización de este proyecto.

Diferenciamos en este caso entre costes materiales y personales:

COSTES MATERIALES		
<u>Concepto</u>	<u>Cantidad</u>	<u>Coste</u>
Dominio .gal	12 meses	16€
Hosting (VPS)	12 meses	36€
	TOTAL	52€
COSTES PERSONALES		
<u>Concepto</u>	<u>Cantidad</u>	<u>Coste</u>
Analista	10 horas	100€
Diseñador	42 horas	336€
Desarrollador Front-end	56 horas	448€
Implantación	2 horas	28€
	TOTAL	912€
TOTAL		
964€		

Para obtener los datos sobre los salarios de los distintos perfiles de informáticos usamos la web indeed.com que aporta datos sobre los salarios medios. De esta forma se aproximó el salario de un analista a 10€/hora, el de un diseñador o desarrollador front-end a 8€/hora y del encargado de la implantación de 14€/hora.

Por otro lado no se tuvieron en cuenta otro tipo de recursos materiales como al disposición de un ordenador o de conexión a internet así como micrófono, cámara para la realización de la presentación.

Capítulo 4: Diseño

En este capítulo se tratarán aspectos relativos al diseño de la aplicación. Tanto hablando de su arquitectura general y bibliotecas y APIs empleadas como al diseño gráfico prestando especial atención en este caso al diseño de las interfaces gráficas.

1. Arquitectura general de la aplicación

En cuanto al diseño general de la aplicación, teniendo en cuenta lo visto en el capítulo anterior, hemos optado por una arquitectura en la que tenemos dos servidores (virtuales): uno que almacena la aplicación web y la base de datos y otro para el sistema de cálculo de rutas.

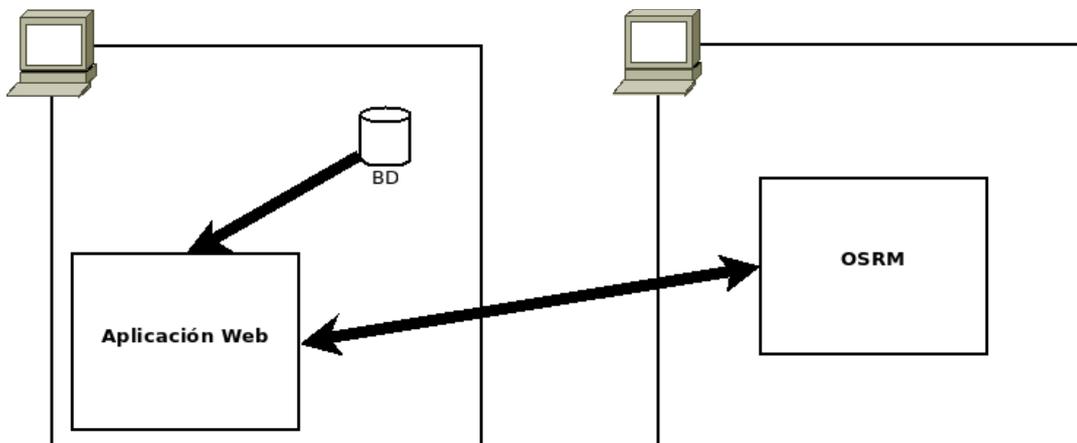


Figura 8: Diagrama de la arquitectura de la aplicación

Como se puede ver en la Figura 8, la base de datos se cargará en la aplicación web que será la encargada de ordenar y filtrar dichos datos atendiendo a diversos criterios. Para ordenar en función del tiempo recorrido en coche se necesitará usar el servidor de cálculo de rutas.

2. Diseño gráfico e interfaces

En esta sección trataremos el diseño gráfico de la aplicación. Para ello trataremos detallaremos la investigación llevada a cabo sobre como se suelen resolver ciertos casos de uso similares para a continuación aportar nuestros prototipos tanto de baja fidelidad como de alta fidelidad.

2.1 Inspiración

En cuanto a la fase de recolección de información, el caso de uso que cubre nuestra aplicación es el de **filtrar y ordenar ciertos elementos**. Este caso de uso es muy visto por webs de venta de productos como Amazon [19] o para web de venta de pisos como Idealista [20]. Para ver como diseñar nuestra aplicación analizamos como los resuelven ciertas circunstancia estas dos webs.

Vemos en el caso de Amazon (Figura 9) que para pantallas grandes y medianas se plantean dos columnas una con los productos y otra con los filtros. Además hay una barra horizontal superior que

muestra el número de productos que se están mostrando junto con un selector para poder cambiar de criterio de ordenación.

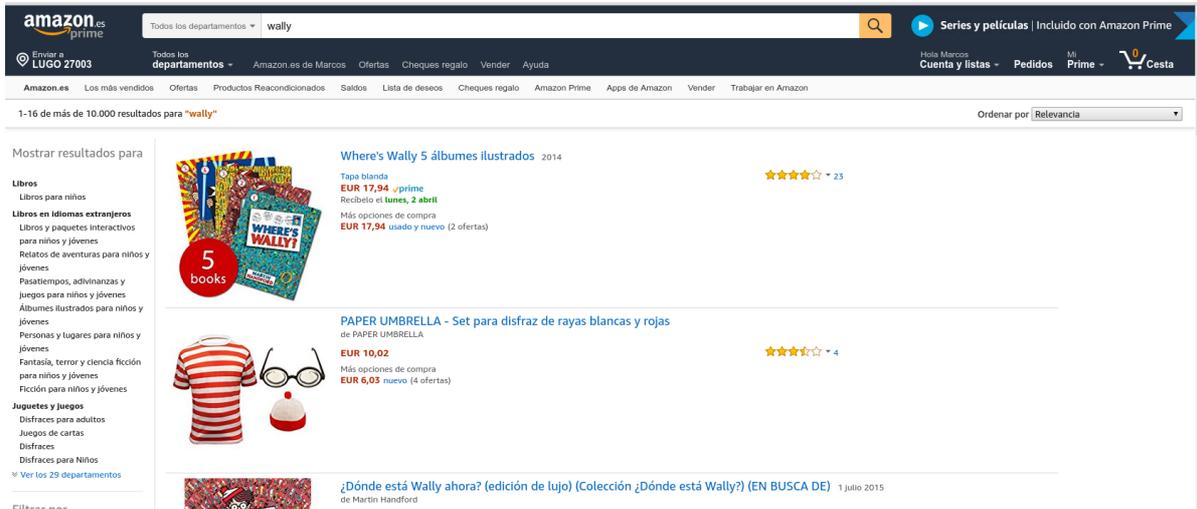


Figura 9: Web Amazon (Escritorio)

Podemos ver como para cada producto se muestran los datos necesarios incluyendo imagen, nombre, categoría y precio entre otros. Esta aproximación es también seguida por webs como El Idealista, como se puede ver en la Figura 10:

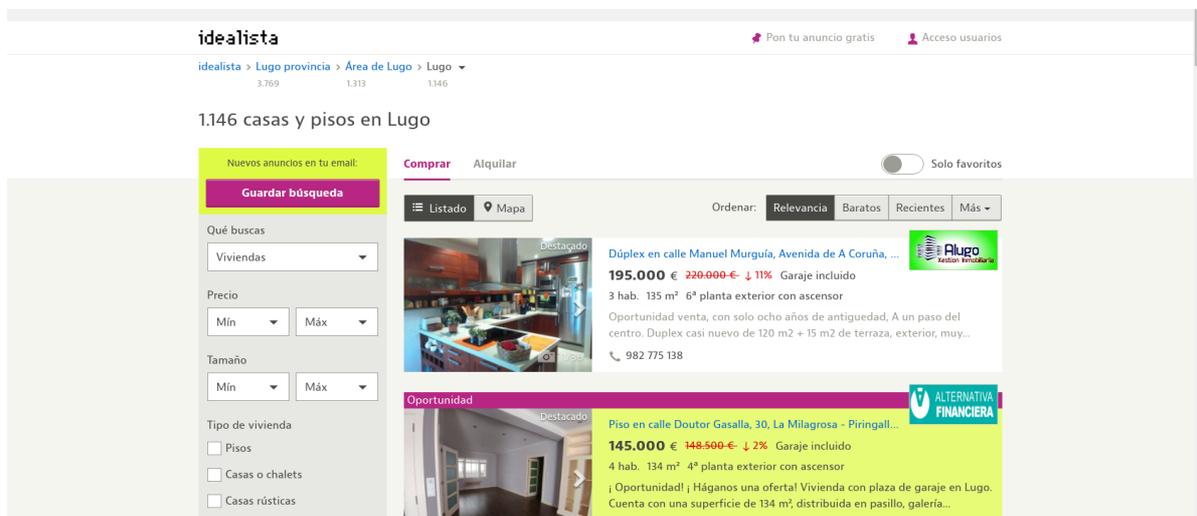


Figura 10: Web Idealista (Escritorio)

Quando bajamos el tamaño de la pantalla veamos como resuelven estas dos compañías la falta de espacio. Podemos ver en el caso de Amazon (Figura 11), como mueve los filtros a un nuevo panel lateral que se muestra pulsando un botón. De esta forma se aprovecha el total del ancho de la pantalla para mostrar los productos y aquello que solo se accede en el para clasificarlos está oculto. Esta solución también es usada en la página web de Idealista.

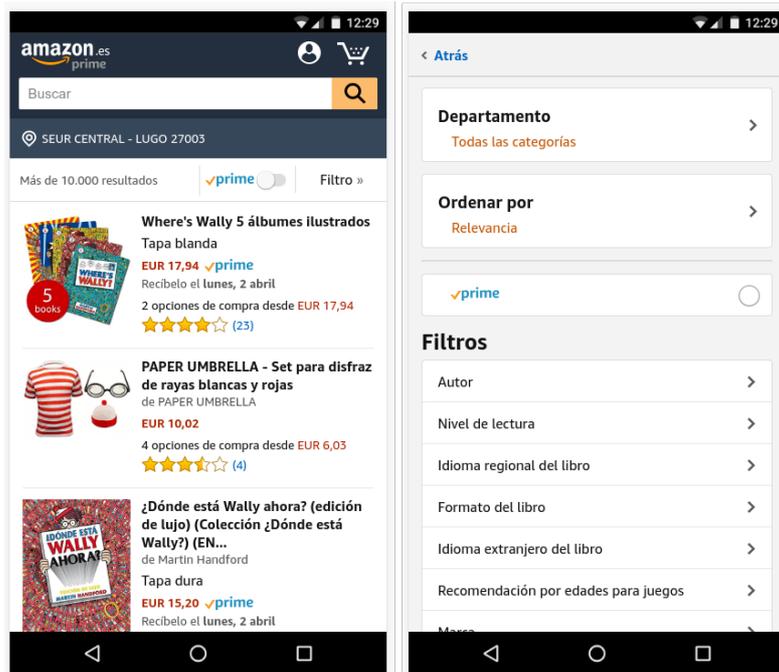


Figura 11: Web Amazon (móvil)

2.2 Prototipos Lo-Fi

Una vez acabada la fase de inspiración, procedemos a realizar diseños de baja fidelidad a mano usando las plantillas provistas por Sneakpeekit [21]. El realizar los diseños a mano alzada nos permite una agilidad que no nos permiten otras herramientas y al mismo tiempo podemos añadir todos los detalles que queramos.

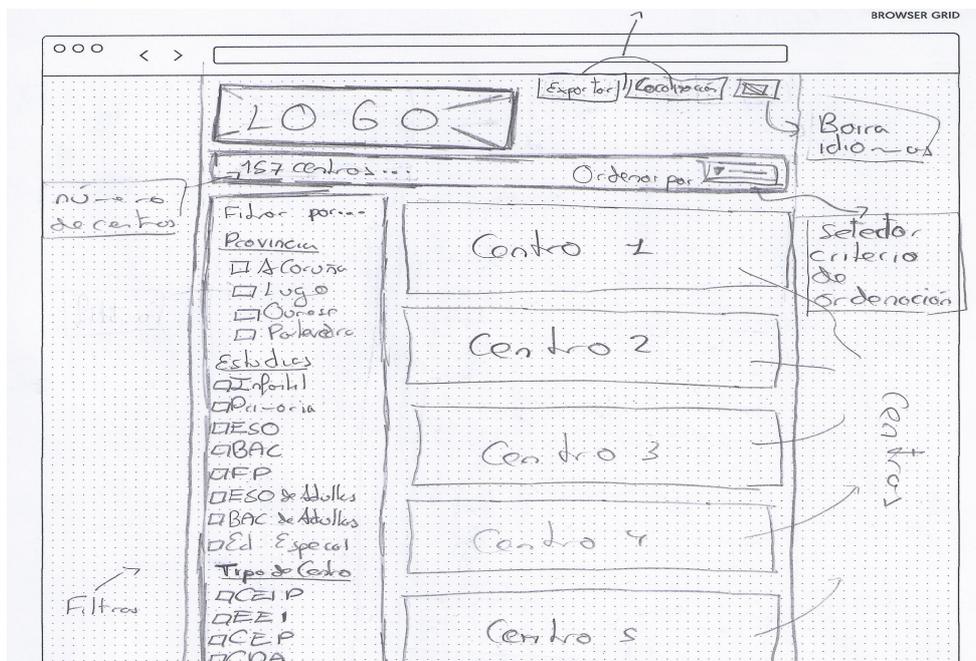


Figura 12: Diseño de baja fidelidad (Escritorio)

En la Figuras 12 y 13 podemos ver el diseño creado para ordenadores de escritorio y móviles. Se puede ver que se ha optado en este caso por una opción muy similar a la planteada por Amazon o el Idealista. Se puede ver, en el caso del diseño para escritorio, dos columnas una de ellas con los filtros y la otra con los centros mientras que en el diseño para móviles esta columna de filtros solo aparecerá al pulsar un botón.

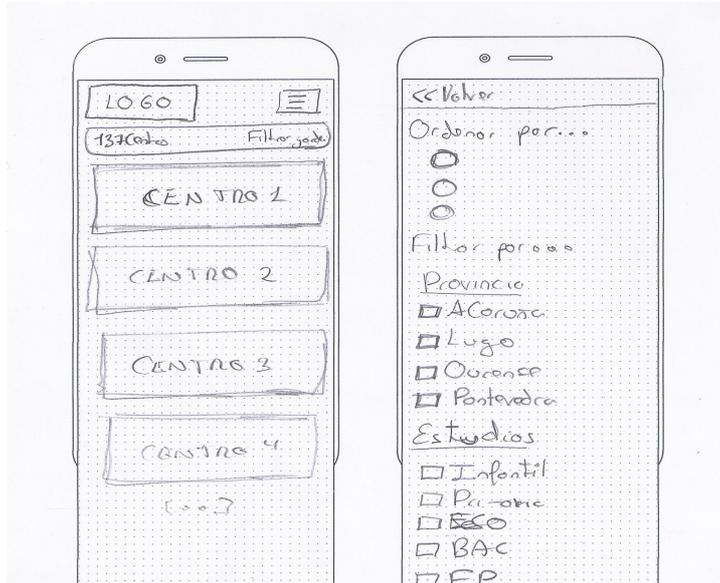


Figura 13: Diseño de baja fidelidad (Móvil)

En cuanto al diseño de los centros en si podemos ver en la Figura 14 como se ha optado por mostrar el nombre del centro, la localidad e información sobre la distancia y el tiempo en coche. Se incluyen además un serie de botones para obtener más datos del centro, eliminar el centro de la lista y copiar el código del centro al portapapeles.

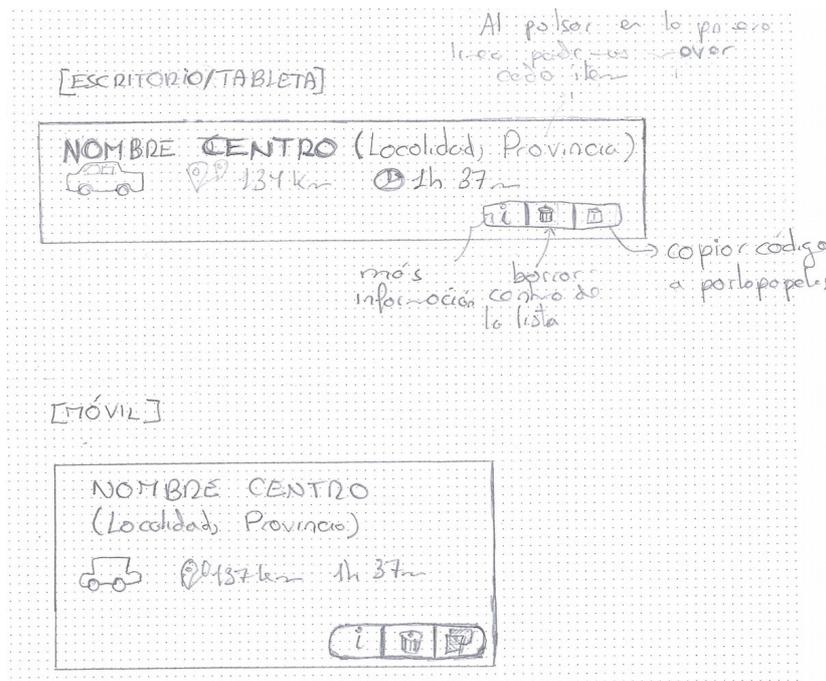


Figura 14: Diseño de baja fidelidad de los centros

Por último, en cuanto a los paneles para cambiar la localización y para exportar los centros se han realizado los diseños que se pueden ver en las Figuras 15 y 16. Se ha optado en este caso por un par de paneles modales que dan la información que necesitamos.

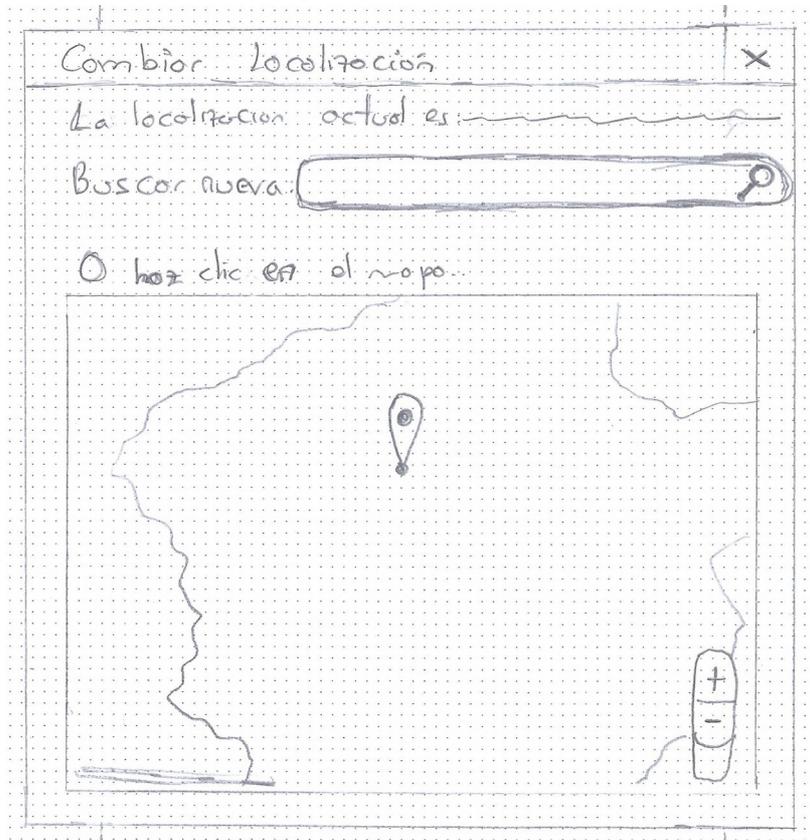


Figura 15: Diseño de panel cambiar de localización

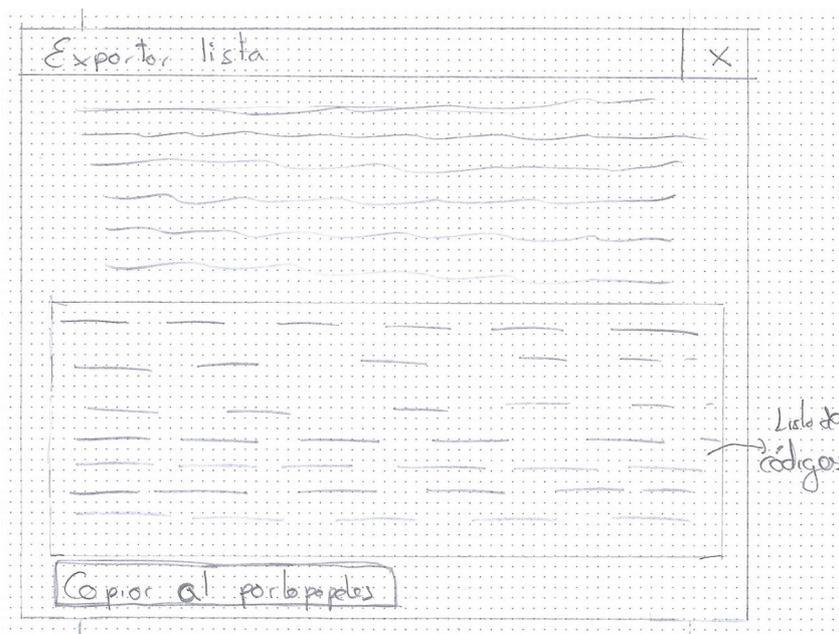


Figura 16: Diseño panel exportar centros

2.3 Prototipos Hi-Fi

Para la realización de los prototipos de alta fidelidad se ha usado código HTML de prueba y el framework CSS Bootstrap por ser este el que, como veremos en la siguiente sección, vamos a usar en el desarrollo final de la aplicación de forma que nos podemos ahorrar cierto tiempo en la implementación.

En la Figura 17 se puede ver el diseño realizado para escritorio. Podemos ver como se ha seguido en su mayor parte el esquema propuesto en el diseño Lo-Fi. EN cuanto a diferencias destacables mencionar el hecho de colocar los botones en vertical debido a que constatamos que se producción un importante ahorro de espacio. También se puede ver como quedaría cada centro cuando se hace clic en el botón de mostrar más información. Por otro lado, se substituye los iconos de banderas para cambiar de idiomas por letras de forma que podemos evitar el uso de banderas como aparece recomendado por el *W3 Consortium* [22].

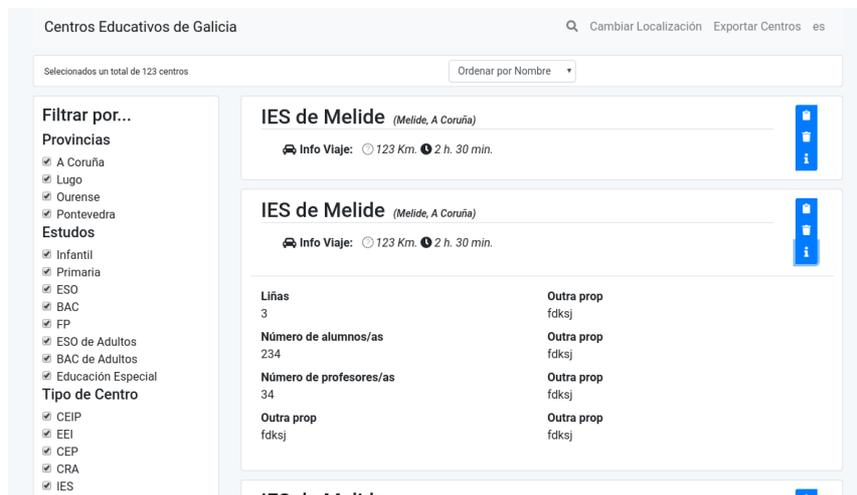


Figura 17: Mockup (Escritorio)

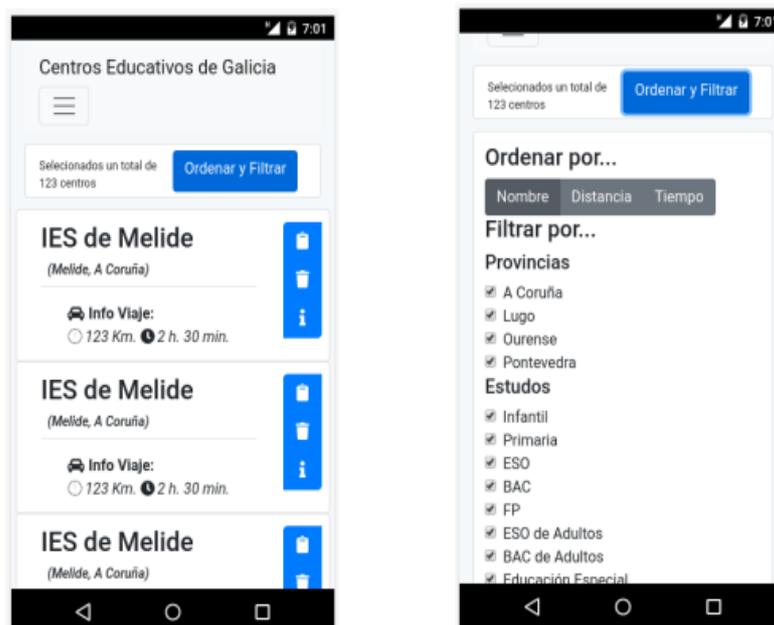


Figura 18: Mockup (Móvil)

Por otro lado, en la Figura 18 se puede ver el diseño para teléfonos móviles.

Vemos que de igual forma se mantiene a grandes rasgos el diseño de baja fidelidad. La única modificación reseñable es que se cambia el panel lateral que muestra los filtros como sucedía en la web de Amazon por un panel que aparece de forma horizontal al pulsar el botón y se contrae al pulsarlo de nuevo.

3. Tecnologías web usadas

Como se describía en la fase de análisis son muchas las tecnologías a usar en la construcción de una aplicación web.

Nuestra aplicación web es una aplicación web estática, con gran cantidad de información pero estática. Es por ello por lo que no tiene sentido usar ninguna tecnología de back-end y se optará por servir el sitio en un simple servidor HTTP (en este caso se usó **NGINX**).

En cuanto al uso de la base de datos consideramos que en aras de simplificar el desarrollo y adaptarlo a los tiempos de este trabajo de fin de máster lo mejor sería no usar ninguna tecnología de bases de datos y servir el conjunto de datos en un fichero JavaScript. Destacar, no obstante, que para extraer los datos de la web de Xunta de Galicia y construir nuestra propia base de datos se usará la biblioteca de análisis web **BeautifulSoup** y el lenguaje **Python**.

Por lo visto anteriormente la mayor parte del peso del trabajo recae en la construcción del front-end. Para esta parte hemos elegido como framework JavaScript **Vue.JS** y como framework CSS **Bootstrap** en conjunción con el uso de SASS.

Se ha optado por Vue.JS debido a ser uno de los más usados actualmente, a que tiene una comunidad de usuarios que desarrollan extensiones bastante importante, que usa JavaScript y a tener una curva de aprendizaje más baja que algunos de sus competidores teniendo la misma potencia.

Además, ha optado por Bootstrap por ser el framework más usado con diferencia y por tener ya ciertos conocimientos sobre su uso, fruto, entre otros, de cursar algunas asignaturas de este máster.

Por otro lado además de estas dos herramientas se ha usado **Webpack** como gestor de dependencias y de tareas, **Atom** como IDE y **Git** como sistema de control de versiones distribuido.

Como servidor de cálculo de rutas hemos elegido **Open Source Routing Machine** por ser Software Libre que podemos implementar en nuestro servidor sin la existencia de limitaciones en cuanto a número de peticiones. Para instalar este servicio en nuestro servidor hemos usado **Docker**.

Capítulo 5: Implementación

En este capítulo se darán detalles sobre aspectos relativos a la implementación del sitio web con los componentes destacados en el capítulo anterior: base de datos, servidor de rutas y front-end.

1. Implementación de la Base de Datos

Como se destacó en el capítulo anterior para simplificar en cierto modo el desarrollo del trabajo de fin de máster así como debido al número de datos no muy elevado se decidió emplear de base de datos un fichero JavaScript con todos los datos de cada centro. El por eso que esta sección versa sobre como se construyó ese fichero.

La página web de referencia de la Xunta de Galicia[4] para obtener información sobre centros educativos permite realizar búsquedas y exportar los resultados en varios formatos como puede ser CSV o XML. No obstante estos resultados que podemos exportar solo contienen aspectos básicos y no proveen toda la información que está disponible en la citada web.

Por ello para extraer toda esta información implementamos un *scraper*, es decir un pequeño programa que analiza el código HTML una web con el objetivo de conseguir la información que hay en el. Para ello usamos el lenguaje de programación Python y la biblioteca BeautifulSoup [23].

```
def get_datos_centro (cod, session):
    ret = {}
    r =
session.get('https://www.edu.xunta.es/centroseducativos/CargarDetalleCentro.do
?codigo=' + cod)
    soup = BeautifulSoup(r.text, 'html.parser')
    ret["nombre"] = soup.find('input', attrs={'name': 'nome'})["value"]
    ret["endereco"] = soup.find('input', attrs={'name': 'endereco.enderezo'})
["value"]
    ret["localidade"] = soup.find('input',
attrs={'name': 'endereco.localidade.nome'})["value"]
    ret["concello"] =soup.find('input', attrs={'name':
'endereco.concello.nome'})["value"]
    ret["cp"] = soup.find('input', attrs={'name': 'endereco.CP'})["value"]
    ret["provincia"] = soup.find('input', attrs={'name':
'endereco.provincia.nome'})["value"]
    ret["telefono"] = soup.find('input', attrs={'name': 'contacto.telefono'})
["value"]
    ret["fax"] = soup.find('input', attrs={'name': 'contacto.fax'})["value"]
    ret["web"] = soup.find('input', attrs={'name': 'contacto.www'})["value"]
    return ret
```

Cuadro 1: Código de función del scraper

Esta página incluye seis pestañas con diferente información de cada centro de las cuales son relevantes para nuestros propósitos tres de ellas. Usando BeautifulSoup capturamos toda esta información y la combinamos con la que ya está en el fichero que se exporta de la página web en un archivo JSON. Por ejemplo la función empleada para capturar la información de la pestaña de

información general se puede ver en el Cuadro 1.

El resultado obtenido para cada centro es de la siguiente forma:

```
{
  coordenadas: {
    lat: 42.891018,
    lon: -8.54292
  },
  ensinanzas: {
    inf: { },
    prim: { }
  },
  cp: 15705,
  endereco: "Avenida BURGO DAS NACIONS s/n",
  web: "http://www.edu.xunta.es/centros/ceipvite1/",
  provincia: "A Coruña",
  nome: "CEIP de Vite I",
  servizos: [
    "transporte"
  ],
  tlf: 981585908,
  xornada: {
    inf: "continua/única",
    prim: "continua/única"
  },
  concello: "Santiago de Compostela",
  cod: 15021883
},
```

Cuadro 2: Ejemplo de centro en la base de datos

Como se puede ver tenemos información sobre la jornada o el tipo de enseñanzas que se imparten el centro que antes no teníamos.

Además hemos substituido los nombres de ciertas enseñanzas por códigos más cortos. Por ejemplo "Educación Infantil" por "inf" para que la información sea más homogénea y podemos crear los filtros de forma más fácil.

2. Implementación del servicio OSRM

En cuanto a la implementación del servicio de cálculo de rutas como se describió en el capítulo anterior, se va a usar el proyecto de OSRM. Existen algunos servicios que proveen esta API pero tienen una limitación de número de peticiones por lo que se optó por alojar este servicio en nuestro servidor usando, para ello usamos la imagen de Docker oficial del proyecto.

Antes de hacer esto debimos descargar los datos de OpenStreetMaps para que el motor de cálculo de rutas funcione con estos datos. Existen ficheros con todos los datos a nivel España pero estos tienen un peso enorme (650MB) y exigen gran cantidad de memoria RAM de la que no disponemos. Por ello y teniendo en cuenta que el ámbito del programa se reduce solo a la Comunidad Autónoma de Galicia, decidimos recortar estos datos usando solo los datos de OpenStreetMaps de Galicia usando la herramienta Osmosis.

A continuación se procesaron estos datos para dejarlos listos para OSRM usando la imagen de docker y se creó el servicio de Systemd que se puede ver en el Cuadro 3 para que este servidor se inicie cada vez que reiniciamos el servidor.

```
[Unit]
Description=OSRM docker instance
Requires=docker.service
After=docker.service

[Service]
Restart=always
ExecStartPre=/usr/bin/docker stop osrm-backend
ExecStartPre=/usr/bin/docker kill osrm-backend
ExecStartPre=/usr/bin/docker rm osrm-backend
ExecStart=/usr/bin/docker run --name osrm-backend -t -i -p 5000:5000 -v
/home/www/osrm/:/data osrm/osrm-backend:v5.12.0 osrm-routed
/data/galicia.osrm

ExecStop=/usr/bin/docker stop -t 2 osrm-backend

[Install]
WantedBy=multi-user.target
```

Cuadro 3: Servicio de Systemd para OSRM

3. Implementación del front-end

En esta sección se tratarán los aspectos relativos a la implementación del front-end siguiendo los diseños realizados. En primer lugar hablaremos del proceso de aprendizaje del framework Vue.js y a continuación trataremos algunos aspectos del desarrollo de la aplicación.

3.1 Aprendizaje del framework Vue.js

Uno de los objetivos que definimos para este trabajo de fin de máster el aprendizaje del framework JavaScript que vamos a usar para este proyecto. Por lo que para realizar este aprendizaje optamos por adquirir un curso de la plataforma Udemy [23] de nombre “Vue JS 2 - The Complete Guide”. Este curso incluye 216 clases en vídeo y un total de 21 horas de las que se visualizaron la mayor parte obviándose los módulos finales por tratar cosas que no se usarán en el proyecto. De esta forma el curso nos aporta formación sobre:

- Qué es VueJS y por qué usarlos.
- Configurando el entorno de desarrollo.
- Aspectos básicos de la sintaxis e interacción con el DOM.
- Uso de componentes.
- Trabajo con formularios.
- Uso de directivas, filtros y mixins.

3.2 Aspectos destacables del desarrollo

De forma genérica se ha usado un componente principal que ha englobado una cabecera, una serie de filtros, una serie de centros y mecanismos de ordenación entre otros. Veremos ahora como se desarrollaron algunos de estos componentes y como se resolvieron ciertas problemáticas surgidas a lo largo del desarrollo de la práctica.

Centros

Al tratarse de una aplicación de ordenar centros, este componente es el más importante. A cada una de las instancias del componente se le pasa por referencia un objeto centro que aporta toda la información que iremos mostrando.

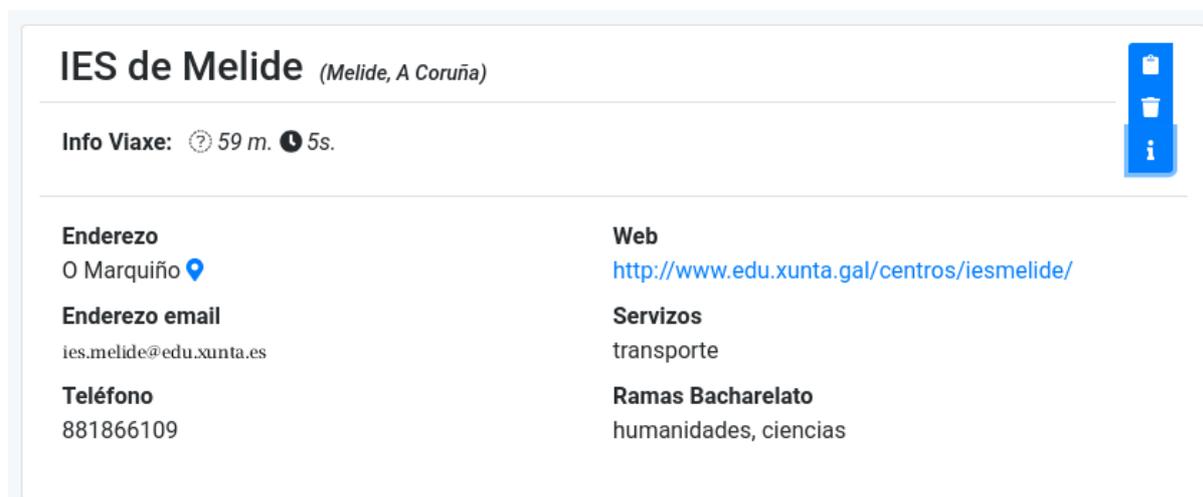


Figura 19: Captura de componente

Si durante el prototipo de alta fidelidad se usaba Bootstrap para implementar el layout de este elemento intentamos evitar el uso de esta librería para este propósito en la implementación final debido a problemas detectados al cargar los *mixins* en los estilos propios del componente. Esta implementación se realizó finalmente usando Flex Box.

Por otro lado la opción de copiar el código se implementó usando el componente *vue-clipboard2* creado por el desarrollador Inndy [25].

Por otro lado, aunque se verá más adelante como se calcula tanto el tiempo como la distancia a los centros, estos datos son dados en segundos y en metros respectivamente por lo que en el componente modificamos el formato antes de mostrar esta información a través de un dato calculado (computed) de Vue.

Veremos más adelante como se implementó la opción de borrar un centro de la lista a través del botón que se incluye en la botonera.

Filtros

Para la implementación de los filtros se pretendía hacer una arquitectura que se expandiese con

facilidad. Por ello se creó por un lado un componente para cada filtro y por otro, un componente que agrupa todos los filtros logrando que se pueda expandir con facilidad.

Los componentes de cada filtro implementan un método de nombre *filter* que indica si un centro determinado cumple con las opciones marcadas en dicho filtro. Además en el momento que se modifican dichas opciones se emite una señal que causa que se recalculen los centros a mostrar.

Por otro lado el componente que agrupa todos los filtros contiene una lista de componentes de tipo filtro. Esta lista se usa por un lado para mostrar todos los filtros y por otro para a la hora de comprobar que un centro se debe mostrar o no tener en cuenta todos los filtros. De esta forma cuando en la el componente principal se quiere filtrar los centro, delegamos en este componente para esta misión. En el Cuadro 4 podemos ver un fragmento de la implementación de este componente.

```
export default {
  data() {
    return {
      activeFilters: [
        'AQDFilterProvincia',
        'AQDFilterTipoCentro',
        'AQDFilterTipoEstudios'
      ]
    };
  },
  components: {
    'AQDFilterProvincia': FilterProvincia,
    'AQDFilterTipoCentro': FilterTipoCentro,
    'AQDFilterTipoEstudios': FilterTipoEstudios
  },
  methods: {
    filter (centers) {
      var i_filter;
      var activeCenters = centers;
      For (var i = 0; i < this.activeFilters.length; i++) {
        i_filter = this.activeFilters[i];
        activeCenters = activeCenters.filter(this.$refs[i_filter][0].filter);
      }
      return activeCenters;
    }
  }
}
```

Cuadro 4: Implementación de componente FilterList

Por otro lado a nivel de estilo, se ha optado por mejorar los selectores creados durante el prototipo que si bien son funcionales no son muy atractivos, por otros personalizados usando Flex Box. En la Figura 20 podemos ver una comparativa entre la implementación realizada durante el prototipo y la realizada durante el desarrollo de la aplicación. No obstante incluimos la versión antigua como fallback ocultándola en los dispositivos que soportan Flex Box.

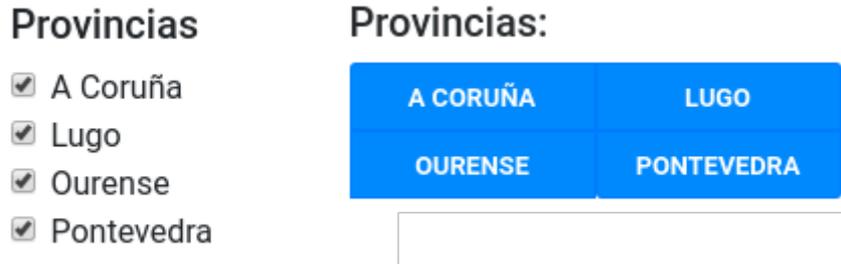


Figura 20: Comparativa filtros en prototipo e en implementación

Mecanismos de ordenación

De igual forma que hicimos en el caso de los filtros también se delegará en un componente propio la misión de ordenar los centros. De esta forma en el componente SortList encontramos una función de nombre *sort* que devuelve una lista de centros ordenada en función del criterio de ordenación marcado.

Se incluyen en este caso tres métodos de ordenación, por distancia, por tiempo y por nombre. Cada método tiene una función de ordenación asociada que es la que se usará en la función *sort* anteriormente citada.

También mejoramos el aspecto del selector del método de ordenación con respecto a lo realizado durante el prototipo de alta fidelidad. El resultado se puede ver en la Figura 22.

Papelera

Para implementar la opción de enviar centros a la papelera, creamos un componente nuevo que por una parte mantendrá una lista actualizada de centros que nos deben mostrar en la lista de centros y por otra nos permitirá ver que centros están en la papelera y poder devolverlos la lista de centros.

Para implementar la opción de borrar un centro en el componente centro, al pulsar el botón se emite una señal que captura el componente papelera y añade el centro en cuestión a la lista de centros en la papelera.

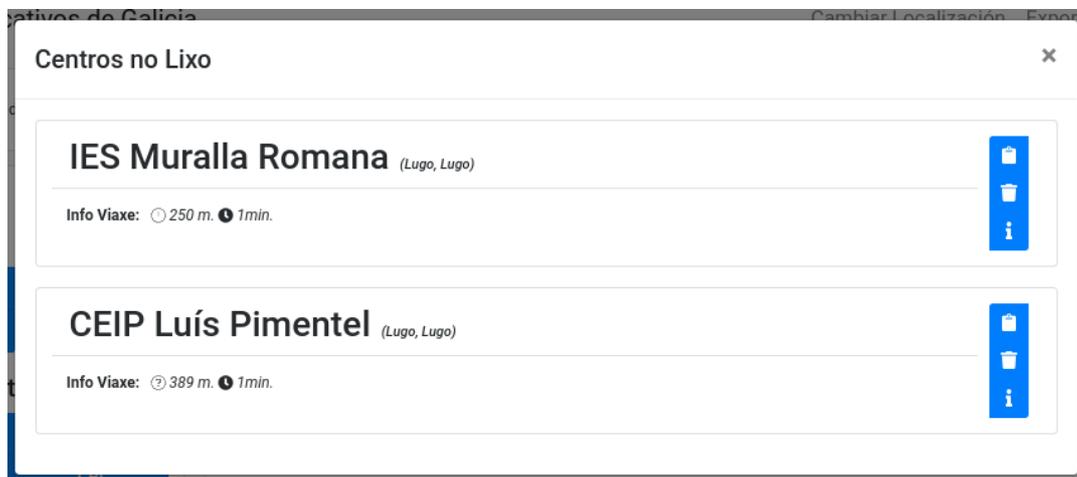


Figura 21: Captura de panel papelera

Por otro, en la interfaz de la aplicación se incluye un contador de centros que está como se puede ver en la Figura 22. Al hacer clic en el botón se abre una ventana modal como la que se puede ver en la Figura 21 que muestra los centros que están en la papelera que volverán a la lista inicial si volvemos a pulsar el botón de la papelera.

Para ocultar los centros en la lista se incluye en el componente papelera una función filtrar que pasándole los centros elimina aquellos que están en la papelera.

Ordenación manual y paginación

Tanto en la prueba de viabilidad como en el prototipo de alta fidelidad se usó la biblioteca SortableJS para implementar la opción de poder ordenar los centros de forma manual. En este caso se usa esta biblioteca de mano del componente creado por el mismo desarrollador *vuedraggable* [26]. Esta biblioteca no solo ordena los centro en la interfaz de la forma que se le pide sino también actualiza la lista de centros con el nuevo orden.

Implementamos además la opción de que cuando cambiamos el orden se pueda volver al orden inicial de forma que también marcamos que el orden de los centros no se corresponde con el original. En la Figura 22 podemos ver como se vería esta opción en la interfaz.

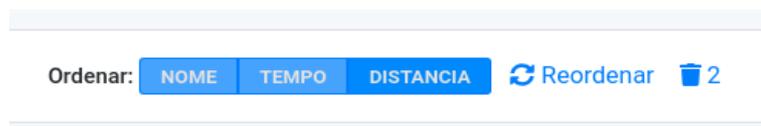


Figura 22: Captura de opciones de ordenar, reordenar y papelera

En cuanto a la paginación, si bien inicialmente no estaba pensado usar paginación para mostrar los centros, detectamos que el framework no era capaz de renderizar todos los centros y la aplicación se realentizaba mucho. En un primer momento se probó usar componentes de terceros pero se detectó que era muy difícil y farragoso hacer compatibles estos componentes con los que ya se estaban usando por lo que hicimos nosotros la paginación a través de una serie de variables y un dato calculado (computed).



Figura 23: Captura de selector de página

A la hora de mostrar las páginas usamos los estilos de Bootstrap como se puede ver en la Figura 23.

Panel de cambiar localización

Para cambiar la localización a tener en cuenta para el cálculo de los tiempos y distancias se incluye una ventana modal que se activa al pulsar la opción específica del menú. En la Figura 24 podemos ver su aspecto.

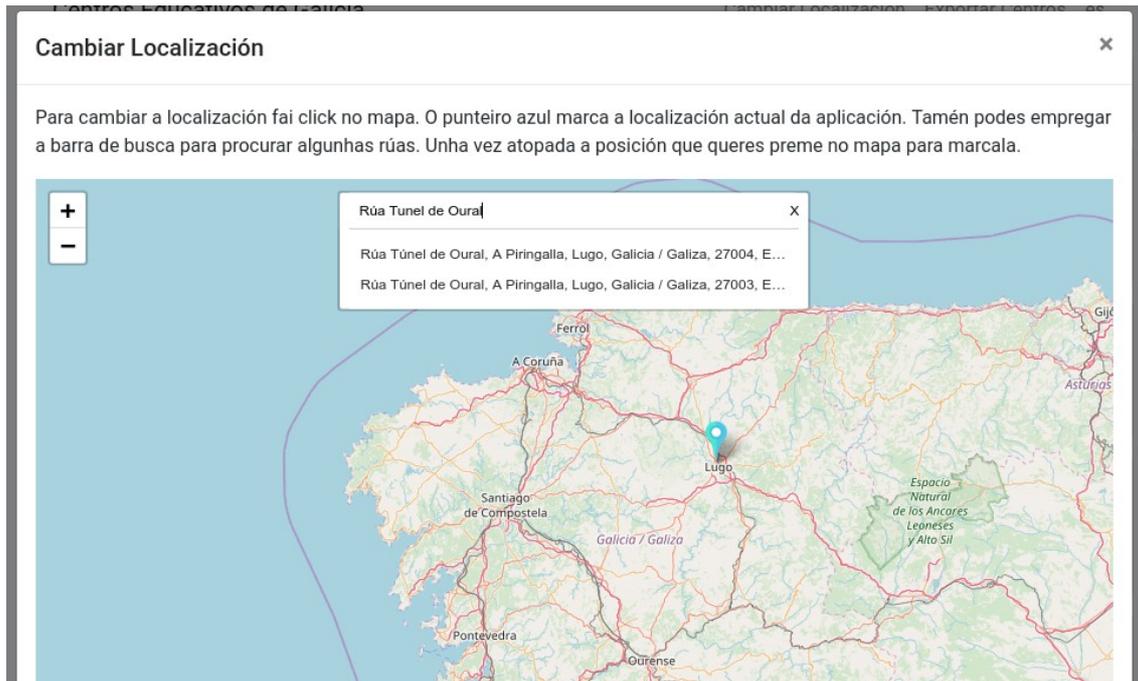


Figura 24: Panel cambiar localización

Para la implementación de esta ventana usamos por una parte la biblioteca Bootstrap para la creación de la ventana modal y por otro la biblioteca Leaflet. Para usar esta biblioteca que permite la creación de planos, usamos un componente creado por el desarrollador KoRiGaN[27] y que nos permite interactuar con la biblioteca de forma sencilla.

Además se incluye el complemento GeoSearch [28] que incluye una barra de búsqueda de calles. De esta forma al pulsar sobre el mapa se cambia la localización actual y se recalculan los tiempos y distancias a cada centros.

Calculo de distancias y tiempos

Por una parte para el cálculo de las distancias se recorren todos los centros en bucle y usando la fórmula de haversine calculamos la distancia en línea recta.

En el caso de los tiempos usamos el servidor OSRM que se ha configurado anteriormente. No obstante como no podemos pedir que se calculen los 1200 centros al mismo tiempo dividimos esta consulta en varias. De esta forma creamos varias peticiones AJAX usando JQuery y las lanzamos todos al mismo tiempo actualizando a continuación la distancia en cada centro. En el Cuadro 5 podemos ver un fragmento del código de esta función.

Estos tiempos se calculan en el momento en el que se modifica la localización y al inicio mientras se están calculando estos dos datos se muestra una advertencia de que se está haciendo esta tarea como se puede ver en la Figura 25.

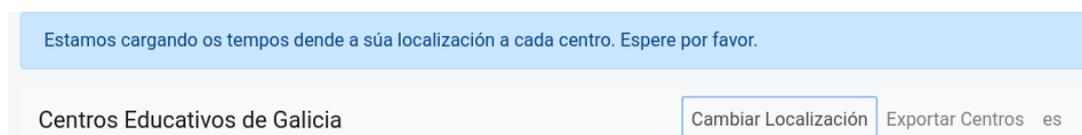


Figura 25: Barra que advierte el cálculo del tiempo a cada centro

```

for (var i = 0; i < centros.length; i += request_max_size) {
  //Get center locations string
  localizaciones = centros.slice(i, i + request_max_size - 1).map(
    function(centro) {
      return centro.coordenadas.lon.toString() + "," +
        centro.coordenadas.lat.toString();
    });

  //Create AJAX call
  ajax_calls.push(
    $.ajax({
      method: "GET",
      url: "https://osrm.aquelando.info/table/v1/driving/" +
        currentLocation.lon.toString() + "," +
        currentLocation.lat.toString() + ";" +
        localizaciones.join(";"),
      data: { sources: 0 }
    })
  );
}

$.when.apply($, ajax_calls).then(
  function() {
    for (var i = 0; i < arguments.length; i++) {
      for(var j = 0; j < arguments[i][0].durations[0].length - 1; j++) {
        index_centro = i*request_max_size + j;
        if (! centros[index_centro].osm) centros[index_centro].osm = {};
        centros[index_centro].osm.tiempo = arguments[i][0].durations[0][j+1];
      }
    }
    callback();
  }
);

```

Cuadro 5: Código función calcular tiempos

Internacionalización y localización

Para llevar a cabo la internacionalización de la aplicación se usó la biblioteca vue-i18n [29]. Esta añade a Vue funciones que a partir de identificadores de cadenas devuelve la cadena traducida en el idioma activo. Estas funciones pueden ser usadas tanto en las plantillas como en el código. En el Cuadro 6 podemos ver un ejemplo de el uso de estas funciones.

```

<div v-if="centro.web.length > 0">
  <dt>{{ $t('web') }}</dt>
  <dd><a href="centro.web">{{centro.web}}</a></dd>
</div>

```

Cuadro 6: Ejemplo de uso de funciones de internacionalización

Por otro lado en lo que respecta a la localización realizamos la localización del programa en Gallego y en Castellano. La citada biblioteca permite añadir las cadenas traducidas a nivel de componente lo cual está muy acorde con la filosofía de componentes de Vue. Estas cadenas se especifican en formato JSON como se puede ver en el Cuadro 7.

```
<i18n>
  {
    "gl": {
      "info-travel": "Info Viaxe:",
      [...]
      "endereço": "Enderezo",
      "servizos": "Servizos"
    },
    "es": {
      "info-travel": "Info Viaje:",
      [...]
      "endereço": "Dirección",
      "servizos": "Servicios"
    }
  }
</i18n>
```

Cuadro 7: Ejemplo de cadenas traducidas

Para realizar el cambio de idioma como estaba planificado en el prototipo de alta fidelidad se añadió una opción en la barra horizontal. En esta opción se muestra los lenguajes disponibles obviando el que está activo y al hacer clic en un lenguaje se cambiaría la localización a este lenguaje.

Capítulo 5: Implantación y pruebas

En este capítulo se tratarán aspectos relativos a la implantación del sitio web y a los ajustes y pruebas realizadas con el sitio.

1. Instalación en el servidor

Una vez realizada la implementación de la aplicación web es el turno de instalarla en el servidor. Se usará un servidor dedicado de la gama Kimsufi[30] de OVH y el nombre de dominio “centroseducativos.gal” contratado con Dinahosting.

Configuramos el nombre de dominio para apuntar tanto “centroseducativos.gal” como “www.centroseducativos.gal” hacia la IP de nuestro servidor.

Se realiza la configuración básica del servidor en cuanto a SSH e instalación de las posibles actualizaciones. A continuación se instala el servidor web NGINX que será el mismo que usaremos como proxy para nuestra instancia de OSRM de Docker.

```
server {
    listen 80;
    server_name centroseducativos.gal www.centroseducativos.gal;
    rewrite ^ https://centroseducativos.gal$request_uri? permanent;
}

server {
    listen 443 ssl;
    server_name www.centroseducativos.gal;
    rewrite ^ https://centroseducativos.gal$request_uri? permanent;
}

server {

    listen 443 ssl;
    server_name centroseducativos.gal;

    ssl_certificate /etc/letsencrypt/live/aquelando.info-0001/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/aquelando.info-0001/privkey.pem;
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_ciphers HIGH:!aNULL:!MD5;
    ssl_stapling on;
    ssl_stapling_verify on;

    add_header Strict-Transport-Security "max-age=31536000";

    access_log /var/log/nginx/centroseducativos.log combined;
    error_log /var/log/nginx/centroseducativos-err.log warn;

    location / {
        root /home/www/centroseducativos/;
        index index.html;
    }
}
```

Cuadro 8: Configuración de VirtualHost NGINX

Para poder usar HTTPS tanto en el servicio OSRM como en el de la propia aplicación, se usarán los certificados proporcionados por Let's Encrypt [31] por lo que instalamos la aplicación *certbot* y solicitamos un certificado para el dominio del servicio de OSRM y para la propia aplicación web.

Una vez configurados los servicios SSL, creamos un VirtualHost apuntando a una carpeta de nuestro equipo como se puede ver en el Cuadro 8. Vemos en dicho cuadro que las primeras directivas son para dirigir todos los accesos que no vayan a la URL canónica a esta.

Para terminar, después de compilar la aplicación usando Webpack, se copian los archivos necesarios en la carpeta establecida.

2. Optimización de velocidad de carga

Una vez instalada la aplicación realizamos un análisis de aquellas cosas en las que se pueda mejorar la velocidad de carga. Para ello usamos la herramienta de Google *PageSpeed Insights* [32] que nos indica que debemos activar la compresión en el servidor para todos los ficheros, minimizar ciertos ficheros, establecer un tiempo de cache para algunos ficheros y optimizar algunas imágenes.

En cuanto a la activación de la compresión modificamos la configuración de Gzip en el servidor NGINX para que se use en todo tipo de archivos. De esta forma descomentamos ciertas líneas para desactivar estas opciones como podemos ver en el Cuadro 9.

```
[...]
##
# Gzip Settings
##

gzip on;
gzip_disable "msie6";
gzip_vary on;
gzip_proxied any;
gzip_comp_level 6;
gzip_buffers 16 8k;
gzip_http_version 1.1;
gzip_types text/plain text/css application/json application/javascript text/xml
application/xml application/xml+rss text/javascript;
[...]
```

Cuadro 9: Activar GZip en NGINX

Usando la herramienta uglify [33] minimizamos el fichero de la base de datos consiguiendo reducir de 690KB a 600KB su tamaño. También empleamos una herramienta online para optimizar algunas imágenes.

En cuanto a establecer el tiempo de cache de algunos ficheros se hace a través del fichero de configuración del VirtualHost creado como se puede ver en el Cuadro 10. De esta forma se establece para las imágenes que son fundamentalmente los correos de cada centro, el tiempo máximo. Por otro lado para los ficheros JavaScript se establece un tiempo de una semana [34].

```
[...]
location ~* \.(png)$ {
    root /home/www/centroseducativos;
    expires max;
    add_header Cache-Control "public";
}

location ~* \.(js)$ {
    root /home/www/centroseducativos;
    expires 7d;
    add_header Cache-Control "public";
}
[...]
```

Cuadro 10: Tiempo de cache para imágenes y ficheros JavaScript

Por último modificamos el script que carga el servicio de análisis web de Google (Google Analytics) para usar una versión más ligera de nombre ga-lite [35].

Con todas estas modificaciones conseguimos muy buena nota en la utilidad de Google (un 88 sobre 100) y muy buena nota en otras páginas como se puede ver en la Figura 26.

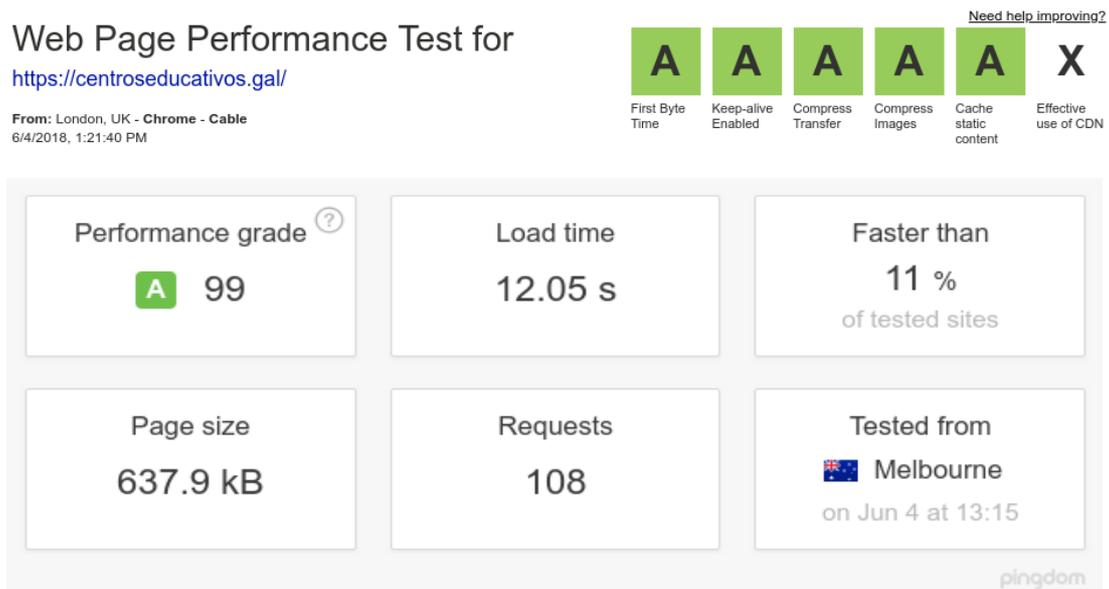


Figura 26: Resultados informes de carga

3. Optimización SEO y promoción

Para analizar el rendimiento en buscadores de nuestra página usamos para ello la utilidad SeoQuake que nos indica todos los *metadatos* que nos faltan para que nuestro sitio esté completo.

Para completar los metadatos de la aplicación se usa el complemento vue-meta [36] que añade los metatags que especifiquemos a la cabecera del documento. Se especifica, título, descripción y además otros metadatos para Facebook y Twitter.

Se registra el sitio en Google y además se crea un sitemap, un favicon y el archivo robots.txt para mejorar la experiencia SEO.

No obstante se detecta que debido a que los metadatos se cargan a través de Vue, Google no detecta bien ni estos metadatos ni el contenido de la página. Para solucionar esto se instala el plugin prerender-spa-plugin [37] que tiene el objetivo de crear un archivo index.html precargado de forma que Google analiza este archivo.

Detectamos por otro lado que esto también mejora en cierto modo nuestra velocidad de carga al eliminar un poco el tiempo de procesamiento de Vue.

Para promocionar el sitio, añadimos un enlace a la prueba de concepto que aun estaba en funcionamiento y con bastantes usuarios y además se comparte este enlace en el grupo de Facebook de Profesores de Galicia [38].

4. Análisis de uso

Para analizar la cantidad de usuarios que entran en nuestra aplicación como se destacó anteriormente, se instala Google Analytics [39]. Tras varios días de uso vemos que la web empieza a tener un importante volumen de usuarios como se puede ver en la Figura 27.

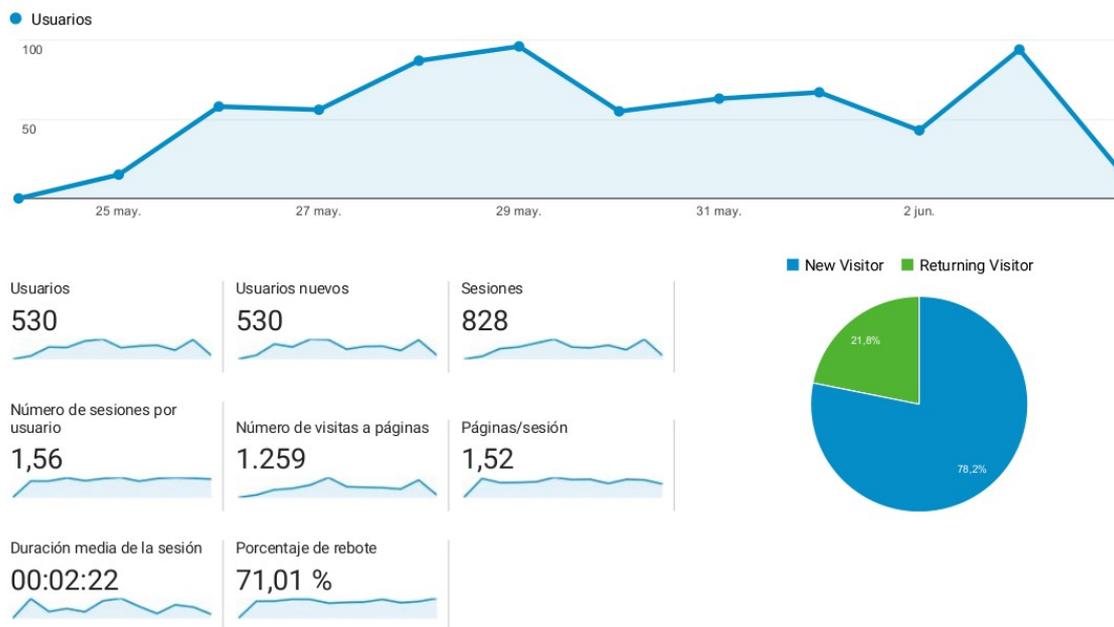


Figura 27: Resultados de la web en Google Analytics

Para afinar más en el uso que le dan nuestros usuarios a la aplicación se creó un evento personalizado que se ejecuta en el momento en el que el usuario exporta la lista de centros. Para realizar esto se usó la API pública de Google Analytics. No obstante este paso fue realizado en los últimos días y aún no se obtuvieron resultados.

Capítulo 6: Conclusiones y trabajo futuro

En este capítulo se detallarán ciertas conclusiones sobre el trabajo realizado y lo aprendido durante este trabajo de fin de carrera que pone fin al máster así como destacar ciertas vías de mejora de la aplicación en un futuro.

1. Conclusiones

El trabajo constituye un colofón en el que usar gran cantidad de aprendizajes que se fueron desarrollando a lo largo del máster. Tanto los conocimientos aplicados en la elección de la metodología, el diseño de la interfaz, las distintas formas de analizar la satisfacción de nuestros usuarios como la gran cantidad de tecnologías que se aplicaron durante este trabajo son fruto en su mayor parte de haber cursado las asignaturas de este máster. Además como punto final hemos aprendido a implementar una aplicación web completa usando un framework como es Vue.js.

Al acabar este trabajo, hemos conseguido completar los objetivos tanto personales como del producto a desarrollar que nos planteamos. Se ha creado una aplicación que si bien es mejorable constituye una herramienta muy buena para los profesores en Galicia. Además hemos aprendido a usar el framework Vue.JS y profundizamos en nuestro conocimiento de Bootstrap sobre todo en lo que se refiere al uso de SASS.

En cuanto a la consecución de los requisitos de la aplicación hemos conseguido cumplir todos si bien sería necesario profundizar en algunos y mejorar el rendimiento de la aplicación como se verá en la sección siguiente.

Vemos, no obstante, que para lograr completar el trabajo con éxito fue necesario el uso de más horas de las que estaban planificadas. Esto es debido a lo complicado que es estimar ciertas tareas como puede ser el aprendizaje de un nuevo lenguaje además de ciertos detalles de la página de la Xunta que no se conocían en un principio.

En cuanto a la metodología, un trabajo con más tiempo sería interesante y muy relevante implementar algún tipo de metodología más rigurosa y más cercana a lo que se usa en la industria y no escoger ciertos detalles de varias metodologías. Se ha intentado no obstante usar de forma intensiva Git de forma que fuéramos documentando los cambios. Y en la actualidad con un producto más maduro seremos capaces de seguir un mejor orden a la hora de hacer las mejoras.

2. Líneas de trabajo futuro

Más allá de la corrección de posibles fallos que se puedan descubrir del uso de la aplicación hay ciertas vías de mejora o expansión que sería interesante explorar en un futuro:

- **Mejorar la fluidez de la aplicación.** Aunque el sitio funciona en ocasiones lo hace bastante lento y los componentes no renderizan los datos después de calcularlos lo que hace que la aplicación de una sensación de lentitud. Sería interesante investigar las causas y buscar

ciertas soluciones que pueden pasar por el uso de una base de datos en vez de un fichero o revisar la implementación de los componentes en búsqueda de algún cuello de botella.

- **Implementar filtros diferentes.** Se han implementado solo los filtros más básicos pero con la información que hemos recabado sobre los centros podríamos filtrar por muchas cosas que sean de utilidad para los profesores y maestros. Algún ejemplo puede ser filtrar dependiendo de las ramas de bachillerato o de los diferentes ciclos de F.P. que se imparten en cada centro.
- **Aportar más información sobre la distancia en carretera o el tipo de vía.** La principal característica de la aplicación que la hace extremadamente útil es la opción de ordenar por tiempo en coche. Es por ello que sería interesante profundizar en esta característica y una forma sería aportar la distancia por carretera real o incluso el tipo de carreteras que se transitarán (dado que no es lo mismo estar 1h o 100km por una autovía y por una carretera nacional). Estos datos se podrían obtener también del servicio OSRM aunque el número de llamadas al API sería mucho mayor.
- **Mejorar la información de los centros.** Por una parte sería interesante seguir recabando información de los centros, por ejemplo diferenciar centros de carácter urbano y rural, centros situados en la costa o en el interior, indicar número de alumnos, de profesores, etc. Toda esta información se podría obtener a partir de cruzar los datos ya disponibles con otras bases de datos así como de hacer que la web sea dinámica permitiendo a los usuarios subir información sobre cada centro.
- **Integración con las aplicaciones de la Xunta de Galicia.** Investigar la forma de integrar esta aplicación con las que provee la Xunta de Galicia para facilitar la participación en los concursos aún mas a los profesores de forma que no tengan que copiar a mano los códigos de los centros una vez ordenados por la aplicación.

Bibliografía

En esta sección se realizará una recopilación de las fuentes bibliográficas empleadas para la redacción de este documento y del Trabajo de Fin de Máster.

- [1] Xunta de Galicia. ORDE do 13 de xuño de 2017 pola que se ditan normas para a adxudicación de destino provisional para o curso académico 2017/18 [...]. [Consultado el 8 de Marzo de 2018] Disponible en: edu.xunta.gal/portal/sites/web/files/20170626_destino_provisional_secundaria.pdf.
- [2] Wikipedia. Fórmula de haversine [Consultado el 20 de marzo de 2018] Disponible en: es.wikipedia.org/wiki/F%C3%B3rmula_del_haversine.
- [3] OSRM Project. OSRM [Consultado el 23 de marzo de 2018] Disponible en <http://project-osrm.org/>
- [4] Xunta de Galicia. Centros educativos. Consulta e mantemento de centros. [Consultado el 23 de marzo de 2018] Disponible en: edu.xunta.es/centroseducativos.
- [5] Junta de Castilla y León. Directorio de centros de Castilla y León. [Consultado el 23 de marzo de 2018] Disponible en: directorio.educa.jcyl.es/es/busqueda-avanzada.
- [6] Gobierno del Principado de Asturias. Directorio de Centros. [Consultado el 23 de marzo de 2018] Disponible en: educastur.es/directorio-de-centros.
- [7] Gobierno de Cantabria. Búsqueda de centros educativos de Cantabria. [Consultado el 23 de marzo de 2018] Disponible en : educantabria.es/centros/busquedacentros.html.
- [8] Gobierno de España. Registro estatal de centros docentes no universitarios. [Consultado el 23 de marzo de 2018] Disponible en: educacion.gob.es/centros.
- [9] WebAssembly. [Consultado el 27 de mayo de 2018] Disponible en: <https://webassembly.org/>.
- [10] Otto, Mark y Thornton, Jacob. Bootstrap. [Consultado el 23 de marzo de 2018] Disponible en: <http://getbootstrap.com/>.
- [11] Zurb. Foundation. The most advanced responsive front-end framework in the world. [Consultado el 25 de mayo de 2018]. Disponible en: <https://foundation.zurb.com/>.
- [12] Semantic. Semantic UI. [Consultado el 25 de mayo de 2018]. Disponible en: <https://semantic-ui.com/>
- [13] Facebook Inc. React. A JavaScript library for building user interfaces. [Consultado el 25 de mayo de 2018]. Disponible en: <https://reactjs.org/>.
- [14] Google. AngularJS. [Consultado el 25 de mayo de 2018]. Disponible en: <https://angularjs.org/>
- [15] You, Evan. Vue.js. The Progressive JavaScript Framework. [Consultado el 23 de marzo de 2018]. Disponible en: <https://vuejs.org>.
- [16] Scrum.org. Scrum. [Consultado el 29 de mayo de 2018]. Disponible en <https://www.scrum.org>.
- [17] Agile Manifesto. [Consultado el 29 de mayo de 2018]. Disponible en <http://agilemanifesto.org>.
- [18] Beck, Kent. Extreme Programming Explained: Embrace Change. [Consultado el 29 de mayo de 2018].
- [19] Amazon.com, Inc. Amazon.[Consultado el 6 de abril de 2018] Disponible en <http://amazon.com>.
- [20] Idealista S.A. idealista — Casas y pisos, alquiler y venta. [Consultado el 6 de abril de 2018] Disponible en <https://www.idealista.com>.

- [21] Vitiello, Pasquale. Sneakpeekit. [Consultado el 5 de abril de 2018] Disponible en <http://sneakpeekit.com/>.
- [22] W3 Consortium <https://www.w3.org/TR/i18n-html-tech-lang/>
- [23] Richardson, Leonard. Beautiful Soup. [Consultado el 20 de abril de 2018] Disponible en <https://www.crummy.com/software/BeautifulSoup>.
- [24] Udemy Inc. Udemy. [Consultado el 5 de mayo de 2018] Disponible en <https://www.udemy.com/>
- [25] Inndy. Vue-clipboard2 [Consultado el 13 de mayo de 2018] Disponible en <https://github.com/Inndy/vue-clipboard2>.
- [26] Desmaisons , David. Vue.Draggable [Consultado el 15 de mayo de 2018] Disponible en <https://github.com/SortableJS/Vue.Draggable>.
- [27] Bouchaud, Mickael. Vue2Leaflet [Consultado el 16 de mayo de 2018] Disponible en <https://github.com/KoRiGaN/Vue2Leaflet>.
- [28] Gutierrez, Fabian. Vue2-leaflet-geosearch. [Consultado el 18 de mayo de 2018] Disponible en <https://github.com/fega/vue2-leaflet-geosearch>.
- [29] Kawaguchi, Kazuya. Vue-i18n. [Consultado el 10 de mayo de 2018] Disponible en <https://github.com/kazupon/vue-i18n>.
- [30] OVH. Kimsufi. [Consultado el 10 de Mayo de 2018]. Disponible en <https://www.kimsufi.com>.
- [31] Internet Security Research Group. Let's encrypt. [Consultado el 10 de mayo de 2018]. Disponible en: <https://letsencrypt.org>.
- [32] Google. PageSpeed Insights [Consultado el 3 de junio de 2018]. Disponible en: <https://developers.google.com/speed/pagespeed/insights/>
- [33] Bazon, Mihai. UglifyJS. [Consultado el 3 de junio de 2018]. Disponible en: <http://isperator.net/uglifyjs/>.
- [34] Digital Ocean. Leverage browser caching for Nginx [Consultado el 3 de junio de 2018]. Disponible en: <https://www.digitalocean.com/community/questions/leverage-browser-caching-for-nginx>.
- [35] Luoto, Jesse. Ga-lite. Small, cacheable and open version of Google Analytics JS client. [Consultado el 3 de Junio de 2018] Disponible en: <https://github.com/jehna/ga-lite>
- [36] De Wet, Declan. Vue-meta. [Consultado el 3 de junio de 2018]. Disponible en: <https://github.com/declandewet/vue-meta>
- [37] Fritz, Chris. Prerender-spa-plugin [Consultado el 3 de junio de 2018]. Disponible en: <https://github.com/chrisfritz/prerender-spa-plugin>
- [38] Opositores/Profesores Matemáticas Galicia. (grupo de Facebook). [Consultado el 20 de mayo de 2018]. Disponible en: <https://www.facebook.com/groups/opositoreseprofesmatesgalicia/>
- [39] Google. Google Analitys [Consultado el 4 de junio de 2018]. Disponible en: <https://analytics.google.com>