



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

# **Creación de una herramienta para la catalogación de empresas, orientada a estudios de gestión**

Trabajo Fin de Grado  
**Grado en Ingeniería Informática**

**Autor:** Paula Tolsá Soler

**Tutor:** Juan Vicente Oltra

**Cotutor:** José Onofre Montesa Andrés

2016/2017



# Resumen

---

Se pretende la creación de una herramienta que permita la clasificación de datos empresariales para posteriores estudios de gestión. Para ello, se ha optado por el desarrollo de un sitio web mediante el uso de las tecnologías PHP, MySQL y JavaScript/JQuery entre otras.

Ofrece una interfaz que permite dar acceso a los datos mediante un conjunto de filtros que el usuario podrá elegir según sus intereses.

Los resultados son mostrados en distintos formatos, tablas y gráficos. Además, con ellos se pueden realizar comparaciones con respecto a los años y obtener ciertos cálculos que favorecen a la obtención de conclusiones.

**Palabras clave:** estadísticas, herramienta web, PHP, MySQL, JavaScript, JQuery, empresa, marketing, segmentación



# Tabla de contenidos

---

Resumen .....	3
Tabla de contenidos .....	5
1. Objetos y objetivos .....	7
2. Introducción .....	7
3. Fuente de datos .....	9
3.1. Instituto Nacional de Estadística(INE) .....	10
4. Desarrollo .....	11
4.1. Análisis y especificación.....	12
4.1.1. Estudio previo.....	12
4.1.2. Características principales de la herramienta .....	14
4.1.3. Comparación con la competencia.....	15
4.2. Diseño .....	16
4.2.1. Diseño de la base de datos.....	16
4.2.2. Diseño de la interfaz gráfica.....	19
4.3. Implementación.....	22
4.3.1. Tecnologías utilizadas .....	23
4.3.1.1. HTML.....	23
4.3.1.2. CSS.....	24
4.3.1.3. JavaScript/JQuery.....	25
4.3.1.4. Bootstrap .....	26
4.3.1.5. MySql.....	26
4.3.2. Entorno empleado.....	27
4.3.2.1. NetBeans .....	27
4.3.2.2. XAMPP .....	28
4.3.3. Librería FusionCharts Suite XT .....	29
4.3.4. Implementación de la herramienta .....	29
4.3.4.1. Archivos php .....	31
4.3.5. Problemas en la implementación.....	67
5. Pruebas .....	69
5.1. Validación CSS .....	69
5.2. Compatibilidad con navegadores .....	70
5.3. Análisis resolución de pantalla .....	70
5.4. Velocidad de carga.....	74
5.5. Correspondencia de datos .....	75
5.6. Pruebas de uso.....	76
6. Conclusiones .....	79
6.1. Posibles mejoras.....	80
7. Anexo .....	81
7.1. Preparación del entorno .....	81
7.2. Importación de la base de datos .....	83
7.3. Manual de la herramienta.....	88
8. Bibliografía.....	93





# 1. Objetos y objetivos

---

El objeto del presente Trabajo de Fin de Grado es la obtención del título de Graduado en Ingeniería Informática, expedido por la Universidad Politécnica de Valencia.

En cuanto a los objetivos, principalmente es la puesta en marcha de una herramienta que ponga a disposición de sus usuarios datos representativos sobre el conjunto de empresas españolas.

Otro objetivo destacable con la construcción de dicha herramienta es facilitar la extracción de conclusiones y proporcionar información para futuros estudios de gestión.

Como objetivos personales están el demostrar la capacidad de poder desarrollar un proyecto desde cero con los conocimientos adquiridos en la titulación y adentrarnos en el mundo del diseño web, mediante la utilización de tecnologías como PHP, HTML, CSS, Bootstrap, JavaScript/JQuery y MySQL.

## 2. Introducción

---

Partiendo de la moraleja “no todo lo que reluce es oro” se ha encaminado el desarrollo de este proyecto. Y es que la sobreexposición informativa en la que nos encontramos dificulta cada vez más la tarea de obtener información de calidad. Así pues, si se desea hacer una búsqueda sobre empresas y sus datos con el fin de realizar estudios, el resultado será una infinidad de información proveniente de distintos sitios de los cuales desconoceremos si muestran datos verídicos o no.

Para resolver este problema, he decidido implementar una herramienta en la que se integren datos reales. La mayoría procedentes del Instituto Nacional de Estadística, mostrados de forma organizada sin necesidad de ir cambiando de medios.

El resultado es una aplicación web con una gran base de datos estadísticos de las empresas a nivel nacional de la que se pueda extraer la información que se necesite mostrándola tal y cómo el usuario prefiera.

Creación de una herramienta para la catalogación de empresas, orientada a estudios de gestión

Así dicha herramienta será mayoritariamente un buscador en el que se podrán ir seleccionando distintos filtros tanto de la información a obtener como de la forma en la que se muestran los resultados.

En el presente documento, se realizará una descripción de las distintas fases por las que se ha pasado hasta la obtención del software, desde el planteamiento y análisis de requisitos, hasta su implementación y diseño de la interfaz visible para el usuario final, incluyendo una explicación de las tecnologías y herramientas empleadas. También será mencionado el proceso de pruebas al que se ha sometido para ser verificado y validado.

Además, se detallarán las principales fuentes de extracción de los datos que han dado lugar a la base de datos desde la que se realizan todas las consultas.

Por último, hablaremos de conclusiones tanto académicas del proyecto como personales. Y, al final del documento, incluiremos la bibliografía que se ha utilizado y un anexo al que se le adjuntará un manual del funcionamiento de la herramienta.



### 3. Fuente de datos

---

Antes de empezar con el desarrollo del proyecto, realicé varias búsquedas para seleccionar las fuentes de donde proveerán los datos.

Como bien he mencionado anteriormente, el resultado fue una infinidad de páginas web. Algunas de ellas, con listados de empresas españolas sobre los que se pueden aplicar filtros para clasificarlas y así calcular los datos estadísticos. Otras, ya con datos estadísticos sobre las empresas. Pero la mayoría con información incompleta y difícil de manipular debido a la diversidad de formatos, sobre todo, extensos documentos con gran parte de información que no es de nuestro interés.

Por tanto, he optado por montar mi propia base de datos con información adaptada al contexto evitando así la redundancia de datos. Otro motivo de esta decisión, es la persistencia de los datos, evitando así la dependencia a fuentes externas que pueden desaparecer o ser alteradas con el tiempo.

Partimos de un conjunto reducido de webs que muestran datos estadísticos fiables y verídicos que se integran con el fin de aplicar sobre ellos una serie de filtros que los clasifiquen. De dichas webs he exportado distinta información que me ha resultado interés a formato .csv y la he importado a mi base de datos MySQL. En alguna ocasión, no se ha podido seguir este procedimiento y los datos han tenido que ser extraídos de documentos e introducidos de forma manual en la base de datos.

La mayor parte de estos datos han sido recogidos del sistema mediante el cual el Instituto Nacional de Estadística(INE) pone a disposición de los españoles los datos estadísticos del territorio conocido como INEbase.

Otra herramienta que se ha empleado para completar la base de datos es una aplicación perteneciente al área de estadísticas del Ministerio de Economía, Industria y Competitividad. Se trata de una aplicación que proporciona información sobre las PYME (micro, pequeña y mediana empresa) elaborada, en gran parte, por Instituto Nacional de Estadística (INE) y publicada en el Directorio Central de Empresas (DIRCE).

### **3.1. Instituto Nacional de Estadística(INE)**

Como se acaba de mencionar, la principal fuente de datos es el Instituto Nacional de Estadística (INE).

El INE es un organismo autónomo de España dedicado a la elaboración de estadísticas oficiales sobre la demografía, economía y sociedad del estado.

Nace con la Ley del 31 de diciembre de 1945 con la misión de recoger, producir y difundir datos y operaciones estadísticas de calidad como censos demográficos y económicos, cuentas nacionales, estadísticas demográficas y sociales, indicadores económicos y sociales, coordinación y mantenimiento de los directorios de empresas, formación del Censo Electoral... Junto con esta información estadística también se encuentra información de carácter institucional y metodológico.

El INE está adscrito al Ministerio de Economía, Industria y Competitividad a través de la Secretaría de Estado de Economía y Apoyo a la Empresa.

Ofrece una web oficial que pone a disposición de todos los miembros de la sociedad, todas y cada una de las actualizaciones que el Sistema estadístico de la Administración del Estado elabora de los distintos campos de estudio.

Para el almacenamiento y distribución de la información que proporciona dicho sitio web, el INE emplea un sistema de base de datos denominado INEbase. Sobre el INEbase se pueden realizar operaciones con el fin de obtener resultados estadísticos que pueden ser utilizados y descargados en distintos formatos.

La información del INEbase está organizada por campos temáticos. Nosotros nos centraremos especialmente en el apartado de economía, el cual da acceso a toda la información relativa a la misma. Concretamente partiremos del Directorio Central de Empresas (DIRCE).

El DIRCE es un producto del INE que proporciona datos sobre la población de empresas españolas. Esta información se actualiza con periodicidad anual y es depurada con frecuencia para conseguir la fiabilidad del usuario. Permite obtener un conjunto de datos estadísticos sobre el número de empresas clasificándolo según número de asalariados, condición jurídica, provincia, comunidad autónoma... Todo ello es posible gracias a fuentes administrativas, estadísticas y privadas como por ejemplo: Censo del Impuesto de Actividades Económicas, Retenciones sobre Rentas del Trabajo Personal, Impuesto sobre el Valor

Añadido, Impuesto de Sociedades e Impuesto sobre la Renta de las Personas Físicas, Registro de Cuentas de Cotización con trabajadores, Registro de trabajadores activos en Cuenta propia, Movimientos del Registro Mercantil, Directorio de actividades económicas del País Vasco, Encuestas estructurales y coyunturales de empresas; encuestas de registro, Inventario de Empresas y Entidades Públicas Ficheros de sectorización de la economía española...

## 4. Desarrollo

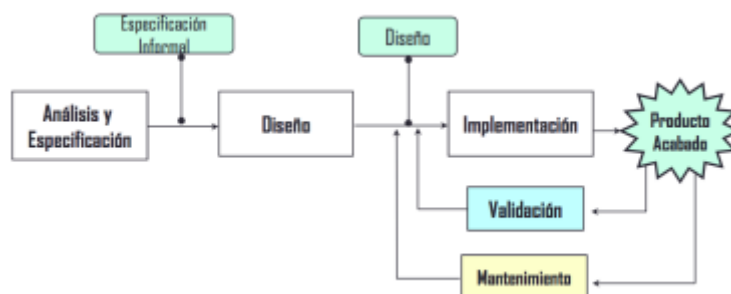
---

En este apartado nos centraremos en el proceso de desarrollo de la herramienta conocido como ciclo de vida software.

El ciclo de vida software es definido según el estándar ISO 12207-1 como: "Un marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso."

Este marco de referencia está formado por cinco etapas intermedias: análisis, diseño, implementación, pruebas y mantenimiento. El orden de la realización de cada una de ellas depende del modelo de ciclo de vida escogido.

En nuestro caso, se ha optado por un desarrollo en cascada o también denominado modelo clásico. Se trata de un enfoque caracterizado por realizar cada etapa del proceso de desarrollo una detrás de otra de tal manera que no se puede avanzar de fase hasta que todas tareas pertinentes dentro de la etapa previa concluyan.



**Figura 1:** Ciclo de vida software clásico o en cascada (Fuente: transparencias asignatura de Ingeniería del Software)

Creación de una herramienta para la catalogación de empresas, orientada a estudios de gestión

El motivo de esta elección es principalmente la sencilla planificación que supone y la calidad del producto obtenido con un equipo de desarrollo que se reduce a una única persona.

A continuación, se explican detalladamente cada una de estas etapas.

## **4.1. Análisis y especificación**

En esta primera fase, comenzaremos haciendo una breve descripción sobre el producto que vamos a construir, cuál será su funcionalidad, cómo se comportará y las necesidades que va a cubrir.

Para ello, previamente se realizará un estudio con el fin de decidir la mejor opción para llevar a cabo la idea abstracta desde la que partimos hacia el producto final.

### **4.1.1. Estudio previo**

El propósito de este proyecto es el de crear una herramienta para la catalogación de empresas. Es decir, una herramienta que muestre datos empresariales clasificados según ciertos criterios.

Son muchas las decisiones que hay que tomar hasta la obtención de este proyecto, pero dos de ellas son fundamentales para poder dar comienzo: qué tipo de herramienta se va a llevar a cabo y cuál es la forma en la que se van a clasificar los datos.

Para cumplir con este propósito, la variedad de implementaciones de la herramienta es infinita, desde unas sencillas hojas de cálculo hasta un complejo programa software.

En mi caso, he optado por desarrollar una web desde cero. Mi elección se basa, principalmente, en poder ampliar conocimientos en el mundo del desarrollo web. Además, es una manera de acercarnos al resultado que se espera gracias al enfoque personalizado que se otorga al desarrollar el proyecto desde cero a partir de unos determinados requisitos y necesidades.

Concretamente, me he decantado por la construcción de una página web dinámica debido al enorme número de ventajas y posibilidades que ofrecen a la hora del desarrollo y el diseño.



Una página web dinámica es una aplicación que proporciona interactividad a los usuarios que la visitan. A diferencia de las páginas web estáticas cuya información es permanente, las dinámicas permiten la posibilidad de mostrar información después de una solicitud hecha por el usuario. Esto es posible al vínculo que mantiene con una base de datos en la que se almacena toda la información deseada sobre la que se pueden hacer distintas acciones. Por tanto, para llevarse a cabo será necesario, por un lado, un panel de administración no visible por los usuarios mediante el que se gestionarán los datos y, por el otro lado, la web públicamente visible para los usuarios.

Entre otras de las ventajas, destacan por la sencillez en el proceso de actualización sin necesidad de entrar en el servidor, pueden realizarse íntegramente con software de libre distribución, existe una amplitud de soluciones predefinidas, permiten la creación de aplicaciones dentro de páginas web...

Con lo que a la forma de clasificar los datos respecta, existe un amplio abanico de posibilidades. Basándome en las utilizadas por las herramientas que he empleado para la obtención de los mismos, he considerado como importantes las siguientes variables de clasificación:

- **Tamaño:** clasificación de la población empresarial según el número de asalariados que componen las empresas. La información obtenida proviene de encuestas realizadas anualmente, de los asalariados inscritos en la Seguridad Social o, en su defecto, mediante la información, si existe, proporcionada por vía fiscal, referida al número de retenciones practicadas en el año. Como modalidades para la clasificación por tamaño se han escogido: sin asalariados, de 1 a 2, de 3 a 5, de 6 a 9, de 10 a 19, de 20 a 49, de 50 a 99, de 100 a 199, de 200 a 499, de 500 a 999, de 1000 a 4999 y de 5000 o más.
- **Condición jurídica:** clasificación de la población empresarial según la contratación jurídica establecida por las empresas. Se obtiene a partir del primer carácter del número de identificación fiscal (NIF). Las modalidades que presenta esta variable son las siguientes: sociedades anónimas, sociedades de responsabilidad limitada, sociedades colectivas, sociedades comanditarias, comunidades de bienes, sociedades cooperativas, asociaciones, organismos autónomos, organizaciones e instituciones religiosas y personas físicas.
- **Sector económico:** clasificación de la población empresarial según el tipo de actividad que realizan las empresas. Distinguimos entre: sector de la industria, sector de la construcción, sector del comercio y resto de servicios.
- **Edad:** según la antigüedad de las empresas. La información es obtenida a partir del periodo en que las unidades son detectadas por primera vez. Las modalidades como referencia para la clasificación por edad que se han



Creación de una herramienta para la catalogación de empresas, orientada a estudios de gestión

escogido son: de 0 a 1 años, de 2 a 3 años, de 4 a 7 años, de 8 a 11 años, de 12 a 15 años, de 16 a 19 años y de 20 o más años.

- Comunidad Autónoma
- Provincia
- Ingresos: clasificación de la población empresarial según la cantidad de beneficios obtenidos en millones de euros. Para catalogarlas, se han considerado como modalidades de referencia: empresas de menos de 2 millones de euros en ingresos, de 2-9 millones, de 10 a 49 millones y con 50 millones o más.

#### **4.1.2. Características principales de la herramienta**

El proyecto surge con la necesidad de tener acceso a datos estadísticos empresariales a nivel nacional desde una única plataforma para facilitar estudios de gestión.

A partir de esta necesidad se han determinado las funciones que podrá realizar la web.

La página web ofrecerá un buscador con un conjunto de filtros que el usuario seleccionará según sus intereses. Los filtros corresponderán con las distintas variables previamente mencionadas. Opcionalmente, se podrá seleccionar una modalidad en concreto del filtro elegido, al igual que también serán seleccionables los años concretos de los que se quiera extraer información. Si no se determina ninguna de estas dos opciones, los resultados corresponderán a todas las combinaciones posibles.

En todos los casos, se obtendrán uno o varios resultados en forma de datos numéricos representando cantidad de empresas del territorio español. Estos datos numéricos podrán ser mostrados o bien en forma de tabla, o bien mediante distintos tipos de gráficas, dejándolo siempre a gusto del usuario.

Como funcionalidad adicional, se podrán realizar comparaciones con respecto a un año base. De los resultados obtenidos, se le dará al usuario la posibilidad de escoger entre uno de los años para que la aplicación calcule las diferencias de dicho año con respecto al resto y poder conocer así, si la evolución en el tiempo ha sido positiva o negativa.

Como se puede ver, los requisitos de una herramienta que cumpla con la descripción anterior son relativamente sencillos. Por este motivo, se ha considerado innecesario entrar en detalle con diagramas que especifiquen los

requisitos dichos, pues solo hay dos casos de uso para un único usuario: buscar información y obtener información.

### **4.1.3. Comparación con la competencia**

A pesar de que actualmente es difícil obtener la información de nuestro interés de forma verídica y organizada, existen dos potentes herramientas con dicho objetivo. Estas son las mencionadas anteriormente como principales fuentes de nuestros datos: el INEbase y la aplicación del Ministerio de Economía, Industria y Competitividad para proporcionar estadísticas. Ambas herramientas ofrecen la misma funcionalidad que nuestra futura herramienta, la obtención de los datos según ciertos criterios.

Al tratarse de páginas webs que manejan y proporcionan mucha información de todo tipo, no solo información del ámbito empresarial, resulta muy costoso navegar por ellas y encontrar lo que se necesita. Además, para el propio navegador también le resulta dificultoso procesar la información lo que ralentiza el proceso de búsqueda incluso llegando a quedar bloqueado. Por tanto, la herramienta desarrollada se caracterizará por su rapidez a la hora de obtener resultados y su facilidad de uso gracias a su interfaz centrada en el usuario.

Otro aspecto a tener en cuenta es que permitirá catalogar las empresas de acuerdo a una única variable, a diferencia del resto que clasifican los datos en función a dos o más a la vez, impidiendo la obtención de resultados concretos.

Las variables que se han utilizado son prácticamente las mismas que las utilizadas en las herramientas existentes, sin embargo, se han añadido la catalogación por sectores y por ingresos. La población de los datos que les corresponden se ha realizado manualmente, a partir de documentos publicados por la web del INE.

Por último, se ofrecerá la funcionalidad adicional de realizar comparaciones de la población de empresas catalogadas de acuerdo a un criterio específico en distintos años de acuerdo a un año base. Así se podrá visualizar la evolución en el tiempo y facilitar estudios de gestión.

## 4.2. Diseño

Una vez conocemos lo que hay que hacer, tenemos que determinar cómo debemos hacerlo. En esta fase, se explicará la estructura interna del producto para su correcto funcionamiento, así como la interfaz de usuario del mismo.

Debido a que se trata de un sitio web con funcionalidades muy concretas, buscar y obtener resultados, no se ha considerado llevar a cabo un desarrollo con arquitectura de tres capas como en la mayoría de desarrollos software. Por este motivo, se ha elegido una lógica de negocio sencilla que divide al proyecto en dos partes bien definidas: por un lado, una capa de presentación con el código necesario para la implementación de la página web y por el otro, la base de datos en la que se almacena toda la información a mostrar.

Para lograr el funcionamiento de la aplicación, ambas partes interactuarán de la siguiente manera: el cliente solicita una información mediante la interfaz web, la capa de presentación interactúa con la base de datos para realizar operaciones de consulta. Los resultados serán devueltos a la capa de presentación mostrándolos al usuario.

### 4.2.1. Diseño de la base de datos

Una base de datos es un conjunto de información perteneciente al mismo contexto que se encuentra estructurada.

A los programas capaces de almacenar y, posteriormente, acceder a los datos de forma rápida se les denominan Sistemas de Gestión de Base de Datos(SGBD). El SGBD escogido es MySQL, un sistema de administración relacional que archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. El acceso a los datos almacenados en dichas tablas se realiza a partir de consultas en lenguaje de consulta estructurado(SQL).

La herramienta que nos permite la administración de MySQL es phpMyAdmin, mediante la cual se han podido crear y modificar las tablas de nuestra base de datos.

Se ha creado una base de datos denominada “TFG” formada por un conjunto de tablas. Cada una de las tablas contiene la información de las empresas de acuerdo a una variable de clasificación, de manera que cuando un usuario seleccione una de ellas, se hará una consulta a la tabla donde se encuentren los datos de las empresas clasificadas por esa variable. Las columnas de las tablas



corresponden con las distintas modalidades de cada variable de clasificación y sus filas a los años a los que pertenecen los datos.

A continuación, se muestra una imagen con la estructura de la base de datos final:

TFG sectoreconomico	TFG ingresos	TFG estratoasalariados
<ul style="list-style-type: none"> <li>Anyo : int(4)</li> <li>Total : decimal(8,1)</li> <li>Industria : decimal(7,1)</li> <li>Construccin : decimal(7,1)</li> <li>Comercio : decimal(7,1)</li> <li>Resto de servicios : decimal(8,1)</li> </ul>	<ul style="list-style-type: none"> <li>Anyo : varchar(4)</li> <li>&lt;2 Millones euros : decimal(8,1)</li> <li>2-9 Millones euros : decimal(6,1)</li> <li>10-49 Millones euros : decimal(6,1)</li> <li>&gt;=50 Millones euros : decimal(5,1)</li> <li>Total : decimal(8,1)</li> </ul>	<ul style="list-style-type: none"> <li>Anyo : int(4)</li> <li>Total : decimal(8,1)</li> <li>Sin asalariados : decimal(8,1)</li> <li>De 1 a 2 : decimal(7,1)</li> <li>De 3 a 5 : decimal(7,1)</li> <li>De 6 a 9 : decimal(7,1)</li> <li>De 10 a 19 : decimal(7,1)</li> <li>De 20 a 49 : decimal(6,1)</li> <li>De 50 a 99 : decimal(6,1)</li> <li>De 100 a 199 : decimal(5,1)</li> <li>De 200 a 499 : decimal(5,1)</li> <li>De 500 a 999 : decimal(5,1)</li> <li>De 1000 a 4999 : decimal(4,1)</li> <li>De 5000 o ms : decimal(4,1)</li> </ul>
TFG edad	TFG condicionjuridica	
<ul style="list-style-type: none"> <li>Anyo : int(4)</li> <li>Total : decimal(8,1)</li> <li>De 0 a 1 años : decimal(7,1)</li> <li>De 2 a 3 años : decimal(7,1)</li> <li>De 4 a 7 años : decimal(7,1)</li> <li>De 8 a 11 años : decimal(7,1)</li> <li>De 12 a 15 años : decimal(7,1)</li> <li>De 16 a 19 años : decimal(7,1)</li> <li>20 o más años : decimal(7,1)</li> </ul>	<ul style="list-style-type: none"> <li>Anyo : int(4)</li> <li>Total : decimal(8,1)</li> <li>Sociedades anónimas : decimal(7,1)</li> <li>Sociedades de responsabilidad limitada : decimal(8,1)</li> <li>Sociedades colectivas : decimal(4,1)</li> <li>Sociedades comanditarias : decimal(4,1)</li> <li>Comunidades de bienes : decimal(7,1)</li> <li>Sociedades cooperativas : decimal(6,1)</li> <li>Asociaciones y otros tipos : decimal(7,1)</li> <li>Organismos autónomos y otros : decimal(5,1)</li> <li>Personas físicas : decimal(8,1)</li> </ul>	

**Figura 2:** Estructura de la base de datos TFG (I) (Fuente: phpMyAdmin)

TFG ccaa	TFG provincias	
# Anyo : int(4)	# Anyo : int(4)	# 24 León : decimal(6,1)
# Nacional : decimal(8,1)	# Nacional : decimal(8,1)	# 25 Lleida : decimal(6,1)
# Andalucía : decimal(7,1)	# 02 Albacete : decimal(6,1)	# 27 Lugo : decimal(6,1)
# Aragón : decimal(6,1)	# 03 Alicante/Alacant : decimal(7,1)	# 28 Madrid : decimal(7,1)
# Asturias, Principado de : decimal(6,1)	# 04 Almería : decimal(6,1)	# 29 Málaga : decimal(7,1)
# Balears, Illes : decimal(6,1)	# 01 Araba/Álava : decimal(6,1)	# 30 Murcia : decimal(7,1)
# Canarias : decimal(7,1)	# 33 Asturias : decimal(6,1)	# 31 Navarra : decimal(6,1)
# Cantabria : decimal(6,1)	# 05 Ávila : decimal(6,1)	# 32 Ourense : decimal(6,1)
# Castilla y León : decimal(7,1)	# 06 Badajoz : decimal(6,1)	# 34 Palencia : decimal(6,1)
# Castilla - La Mancha : decimal(7,1)	# 07 Balears, Illes : decimal(6,1)	# 35 Palmas, Las : decimal(6,1)
# Cataluña : decimal(7,1)	# 08 Barcelona : decimal(7,1)	# 36 Pontevedra : decimal(6,1)
# Comunitat Valenciana : decimal(7,1)	# 48 Bizkaia : decimal(6,1)	# 26 Rioja, La : decimal(6,1)
# Extremadura : decimal(6,1)	# 09 Burgos : decimal(6,1)	# 37 Salamanca : decimal(6,1)
# Galicia : decimal(7,1)	# 10 Cáceres : decimal(6,1)	# 38 Santa Cruz de Tenerife : decimal(6,1)
# Madrid, Comunidad de : decimal(7,1)	# 11 Cádiz : decimal(6,1)	# 40 Segovia : decimal(6,1)
# Murcia, Región de : decimal(7,1)	# 39 Cantabria : decimal(6,1)	# 41 Sevilla : decimal(7,1)
# Navarra, Comunidad Foral de : decimal(6,1)	# 12 Castellón/Castelló : decimal(6,1)	# 42 Soria : decimal(5,1)
# País Vasco : decimal(7,1)	# 13 Ciudad Real : decimal(6,1)	# 43 Tarragona : decimal(6,1)
# Rioja, La : decimal(6,1)	# 14 Córdoba : decimal(6,1)	# 44 Teruel : decimal(5,1)
# Ceuta : decimal(5,1)	# 15 Coruña, A : decimal(6,1)	# 45 Toledo : decimal(6,1)
# Melilla : varchar(10)	# 16 Cuenca : decimal(6,1)	# 46 Valencia/València : decimal(7,1)
	# 20 Gipuzkoa : decimal(6,1)	# 47 Valladolid : decimal(6,1)
	# 17 Girona : decimal(6,1)	# 49 Zamora : decimal(6,1)
	# 18 Granada : decimal(6,1)	# 50 Zaragoza : decimal(6,1)
	# 19 Guadalajara : decimal(6,1)	# 51 Ceuta : decimal(5,1)
	# 21 Huelva : decimal(6,1)	# 52 Melilla : decimal(5,1)
	# 22 Huesca : decimal(6,1)	
	# 23 Jaén : decimal(6,1)	
	# 24 León : decimal(6,1)	

**Figura 3:** Estructura de la base de datos TFG (II) (Fuente: phpMyAdmin)

Para entender mejor la estructura de nuestra base de datos, se propone un ejemplo de una de las tablas poblada como puede ser la tabla correspondiente a la variable de catalogación “estrato de asalariados/tamaño”:

Anyo	Total	Sin asalariados	De 1 a 2	De 3 a 5	De 6 a 9	De 10 a 19	De 20 a 49	De 50 a 99	De 100 a 199	De 200 a 499	De 500 a 999	De 1000 a 4999	De 5000 o ms
2001	2645317.0	1408792.0	892964.0	259764.0	126050.0	86308.0	48205.0	12953.0	5867.0	3180.0	744.0	418.0	72.0
2002	2710400.0	1425332.0	716914.0	271764.0	131666.0	89840.0	50595.0	13468.0	6190.0	3276.0	815.0	466.0	74.0
2003	2813159.0	1459938.0	760515.0	285651.0	136679.0	93823.0	51619.0	13486.0	6402.0	3540.0	878.0	550.0	78.0
2004	2942583.0	1500396.0	818149.0	303783.0	143417.0	98245.0	53267.0	13470.0	6650.0	3590.0	945.0	592.0	79.0
2005	3064129.0	1574166.0	850367.0	310787.0	145443.0	100211.0	56260.0	14394.0	7003.0	3817.0	967.0	630.0	84.0
2006	3174393.0	1616883.0	881748.0	328820.0	154635.0	105470.0	58725.0	15087.0	7367.0	3908.0	999.0	658.0	93.0
2007	3336657.0	1706140.0	932526.0	340638.0	158159.0	108866.0	60738.0	15737.0	7780.0	4218.0	1072.0	684.0	99.0
2008	3422239.0	1754374.0	958711.0	345848.0	160460.0	110369.0	61709.0	16242.0	8061.0	4511.0	1117.0	725.0	112.0
2009	3355830.0	1767470.0	919092.0	332671.0	151233.0	101601.0	55641.0	15075.0	7672.0	3578.0	1008.0	678.0	111.0
2010	3291263.0	1774005.0	893005.0	318155.0	143016.0	88396.0	48765.0	13745.0	7098.0	3374.0	952.0	651.0	101.0
2011	3250576.0	1795321.0	847952.0	313408.0	138040.0	84834.0	46160.0	13279.0	6585.0	3288.0	923.0	679.0	107.0
2012	3199617.0	1764987.0	867550.0	288896.0	131944.0	79113.0	43070.0	12700.0	6434.0	3214.0	935.0	669.0	105.0
2013	3146570.0	1681588.0	922646.0	280643.0	125029.0	74204.0	39506.0	11745.0	6130.0	3450.0	885.0	643.0	101.0
2014	3119310.0	1672483.0	921000.0	284612.0	110619.0	70226.0	38157.0	11223.0	5753.0	3346.0	942.0	646.0	103.0
2015	3186878.0	1754002.0	899802.0	287430.0	112527.0	71518.0	39101.0	11503.0	5928.0	3340.0	937.0	683.0	107.0
2016	3236582.0	1791909.0	895574.0	292403.0	117293.0	75022.0	40895.0	12020.0	6243.0	3489.0	951.0	674.0	109.0

**Figura 4:** Ejemplo de una tabla de la base de datos. Tabla estratoasalariados (Fuente: phpMyAdmin)

#### 4.2.2. Diseño de la interfaz gráfica

Antes de pasar al desarrollo de la web, se realizó un posible diseño de la interfaz gráfica.

La interfaz de usuario gráfica (IGU) es el conjunto de elementos gráficos que hacen posible la interacción del usuario con el sistema informático. Por este motivo, el diseño de una buena interfaz web es un requisito fundamental para la obtención de resultados de calidad. Para ello, se han seguido una serie de principios de diseño y guías de estilo con el objetivo de construir una plataforma atractiva visualmente, sencilla de manejar e intuitiva para los usuarios.

Además, es muy importante la organización del contenido. Se ha decidido realizar dos páginas principales, una correspondiente a la búsqueda de datos y otra para mostrar los resultados. Ambas siguen el mismo patrón cuya estructura es la siguiente:

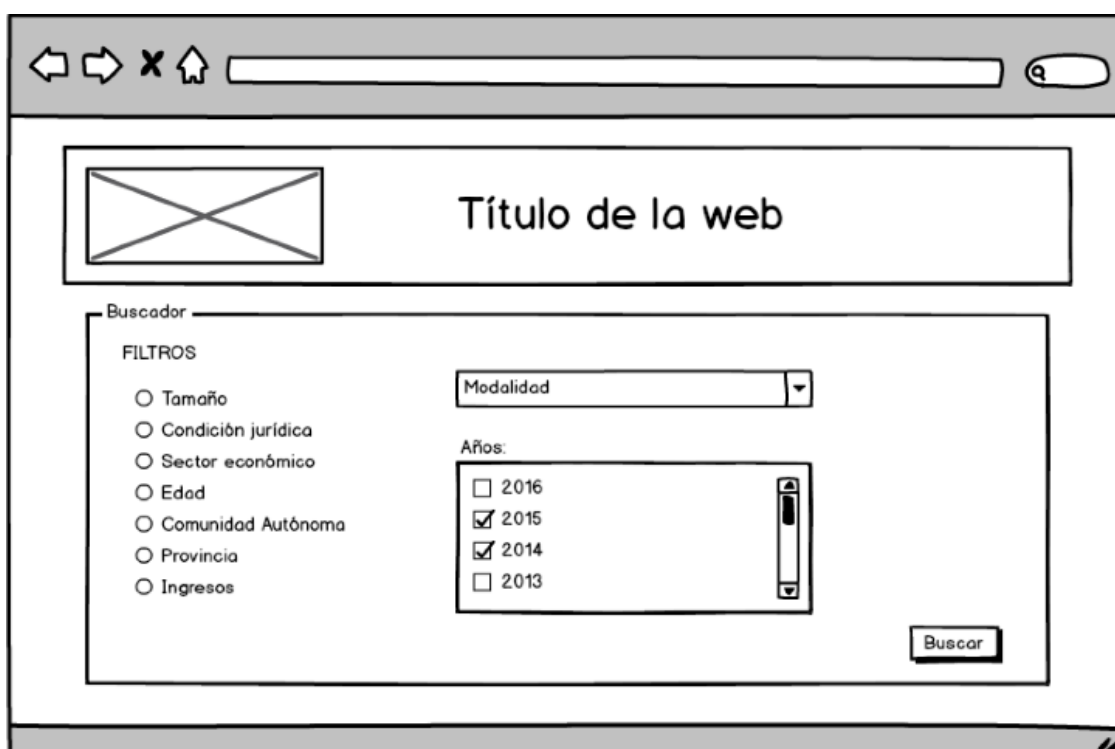
- Una cabecera con el título de la web con una imagen de fondo representativa.
- El cuerpo central formado por un contenedor que varía en función de la página en la que nos encontremos.



- Un pie de página

Para plasmar todas estas ideas, se ha utilizado una herramienta de prototipado de alta fidelidad como es Balsamiq Mockups. Esta herramienta nos ha facilitado la construcción de cada una de las posibles ventanas que formarán la web que, a pesar de que el resultado obtenido no tiene por qué seguir al pie de la letra estos bocetos, resultará de gran ayuda para comenzar con el desarrollo.

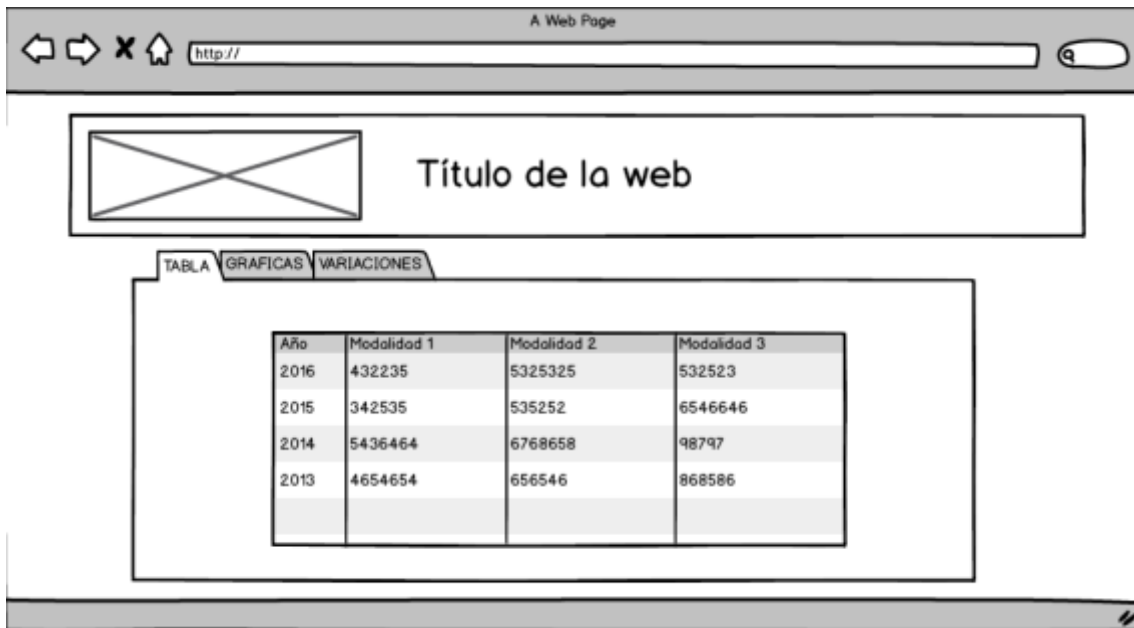
Con lo que respecta a la pantalla principal, la que el usuario se encontrará al acceder a la web y mediante la cual podrá hacer búsquedas personalizadas:



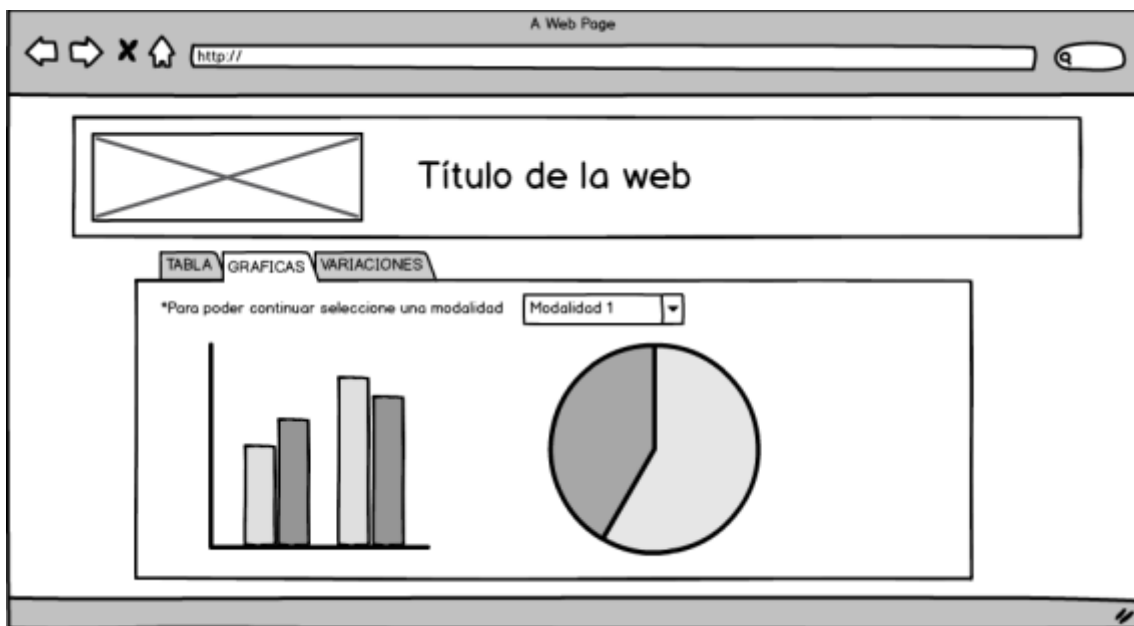
**Figura 5:** Prototipo de la pantalla inicial realizado con Balsamiq Mockups  
(Fuente: Balsamiq Mockups)

La siguiente pantalla que aparecerá al accionar sobre el botón buscar, corresponderá con la ventana que muestra los resultados en función de la búsqueda. Debido a la posibilidad de formatear dichos resultados en una tabla o en gráficas y a la funcionalidad adicional de realizar comparaciones, estará compuesta por tres pestañas de tal forma que estando en la misma pantalla y dependiendo de la que se seleccione, el contenido cambiará dinámicamente.

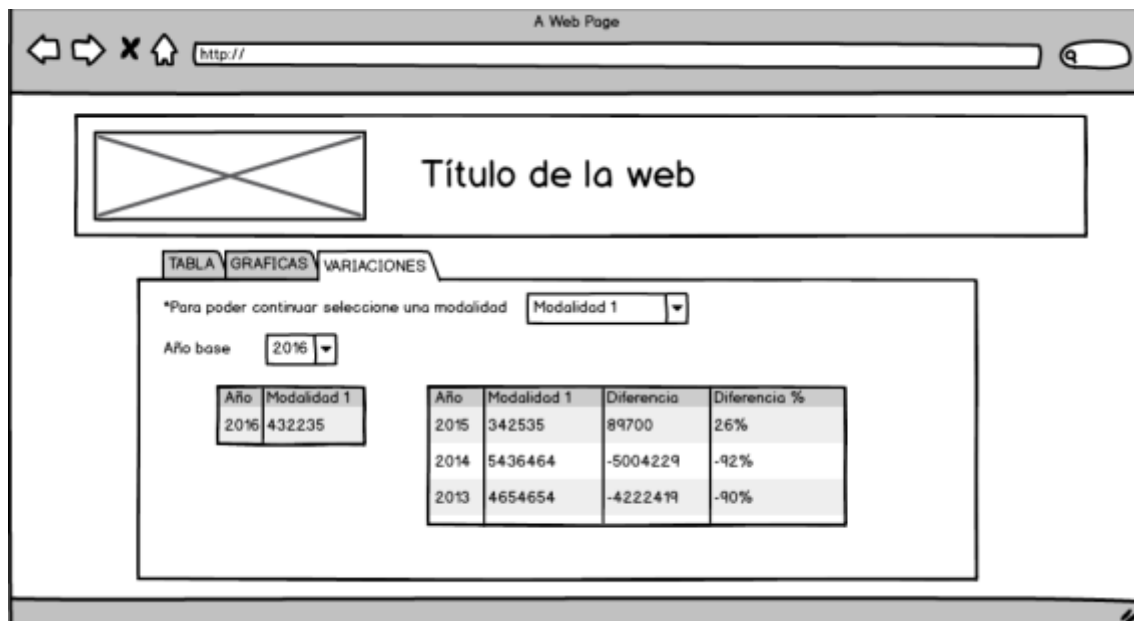
Su apariencia será similar a la mostrada a continuación:



**Figura 6:** Prototipo de la página de resultados. Pestaña TABLA (Fuente: Balsamiq Mockups)



**Figura 7:** Prototipo de la página de resultados. Pestaña GRAFICAS TABLA (Fuente: Balsamiq Mockups)



**Figura 8:** Prototipo de la página de resultados. Pestaña VARIACIONES  
(Fuente: Balsamiq Mockups)

La última tarea de esta fase ha sido buscar un título que identifique a la web. Tras varias opciones, se ha optado por “*Business cataloging*”. Se deseaba un nombre que con pocas palabras describiera el concepto de la herramienta y a su vez resultara atractivo y sencillo de recordar.

### 4.3. Implementación

El producto de esta etapa será el código basado en las especificaciones y diseño anteriormente establecidos.

En primer lugar, se definirán las decisiones de implementación tomadas a partir del estudio previo, así como tecnologías, librerías y entorno empleados. Posteriormente, se describirán los detalles del desarrollo de la capa de presentación que desempeñará la funcionalidad que se vio al plantear el proyecto.

### **4.3.1. Tecnologías utilizadas**

Como bien se ha mencionado anteriormente, la herramienta es un sitio web. Por ello, para su construcción partiremos del lenguaje estándar web HTML 5.0 junto con el uso de hojas de estilo CSS para dar formato. Además, al ser dinámica haremos uso del lenguaje PHP, para mostrar el contenido dinámico, y JavaScript, concretamente su librería JQuery, para llevar a cabo las operaciones de interacción con el usuario. Todo esto requiere la disposición de un servidor hosting (alojamiento web) que permita usar una base de datos MySQL y que permita y soporte PHP. Pues los datos de las secciones dinámicas de la web se guardarán en la base de datos y con PHP accederemos a estos datos para mostrarlos. El servidor utilizado es Apache proporcionado por el entorno XAMPP.

Todas estas tecnologías mencionadas han sido escogidas, principalmente, gracias a la forma en la que se complementan unas con las otras para lograr una aplicación web íntegra, eficiente y de calidad. Además, son de distribución libre, multiplataforma, de fácil despliegue y su uso en la actualidad hace que exista una gran comunidad que da soporte y propone soluciones, lo cual facilita la tarea de programación.

A continuación, van a ser detalladas.

#### **4.3.1.1. HTML**



HTML es un lenguaje de marcado descriptivo que se utiliza para el desarrollo de páginas web.

Es un estándar que define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web.

Se considera el lenguaje web más importante siendo su invención crucial en la aparición, desarrollo y expansión de la World Wide Web (WWW).

Para desarrollar una página web en HTML es necesario crear un documento HTML. Básicamente un documento HTML es un archivo de texto que tienen la extensión .html o .htm. Este documento de texto está compuesto por una serie de etiquetas que el navegador interpreta y da forma en la pantalla. HTML dispone de etiquetas para imágenes, hipervínculos que nos permiten dirigirnos a otras páginas, saltos de línea, listas, tablas, etc.

Consta de varios componentes vitales, los elementos y sus atributos, tipos de datos y la declaración de tipo de documento. Pero no es un lenguaje de programación de por sí, ya que no cuenta con funciones aritméticas, variables o estructuras de control propias de los lenguajes de programación, por lo que HTML genera únicamente páginas web estáticas, sin embargo, HTML se puede usar en conjunto con diversos lenguajes de programación para la creación de páginas web dinámicas.

#### 4.3.1.2. CSS



Las hojas de estilo en cascada (o CSS) es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado como es HTML o en documentos XML.

CSS tiene como objetivo principal separar el contenido de la presentación. Esta separación busca mejorar la accesibilidad del documento, proveer más flexibilidad y control en la especificación de características presentacionales, permitir que varios documentos HTML compartan un mismo estilo usando una sola hoja de estilos separada en un archivo .css, y reducir la complejidad y la repetición de código en la estructura del documento.

Funciona a base de reglas, es decir, declaraciones sobre el estilo de uno o más elementos. Las hojas de estilo están compuestas por una o más de esas reglas aplicadas a un documento HTML o XML. La regla tiene dos partes: un selector y la declaración. A su vez la declaración está compuesta por una propiedad y el valor que se le asigne.



La especificación CSS describe un esquema prioritario para determinar que reglas de estilo se aplican si más de una regla coincide para un elemento en particular. Estas reglas, aplicadas con un sistema llamado en cascadas, las prioridades son calculadas y asignadas a las reglas, así que los resultados son predecibles.

La especificación CSS es mantenida por el World Wide Web Consortium (W3C).

#### 4.3.1.3. JavaScript/JQuery



JavaScript (a veces abreviado como JS) es un lenguaje ligero e interpretado, orientado a objetos con funciones de primera clase, pero también usado en muchos entornos sin navegador, tales como node.js o Apache TouchDB. Es un lenguaje script multi-paradigma, basado en prototipos, dinámico, soporta estilos de programación funcional, orientada a objetos e imperativa.

Es un dialecto del estándar ECMAScript, por este motivo, todos los navegadores modernos interpretan el código JavaScript.

Surgió con el objetivo inicial de programar ciertos comportamientos sobre las páginas web, respondiendo a la interacción del usuario y la realización de automatismo sencillos, por eso, se utiliza principalmente en el lado del cliente (*client-side*), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas. Aunque actualmente es ampliamente utilizado para enviar y recibir información del servidor junto con ayuda de otras tecnologías como AJAX.

Es también significativo su uso en aplicaciones externas, como es en los documentos PDF o aplicaciones de escritorio.

Con él, el programador es capaz de alterar cualquier cosa que se muestra en una página, cambiando, insertando o eliminando todo tipo de contenido. Si lo deseas, puedes controlar de cada cosa que ocurre en la página cuando la está visualizando el usuario y comunicar con él con todo tipo de interfaces especiales.

Tiene tipos y operadores, objetos básicos y métodos. Su sintaxis viene de los lenguajes Java y C, por lo que muchas de las estructuras de esos lenguajes se aplican también a JavaScript. Una de las principales diferencias es que JavaScript no tiene clases.

JQuery es una librería de código JavaScript caracterizada por su facilidad de uso y su gran potencia. Ha sido optimizado para realizar muchas funciones de script frecuentes y lo hace a la vez que utiliza menos líneas de código.

#### 4.3.1.4. Bootstrap



Bootstrap es un framework o conjunto de herramientas de código abierto para el diseño de sitios y aplicaciones web. Ha sido desarrollado y liberado por Twitter.

Facilita y agiliza el desarrollo web front-end a través de plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos basados en HTML y CSS, así como, extensiones de JavaScript. El código fuente de estas plantillas utiliza dos preprocesadores: Less y Sass.

Está orientado a todo el público, por ello, existen varias formas de ser usado. La más sencilla es enlazar la hoja de estilo bootstrap, previamente descargada, con el documento HTML. Otra opción es compilar el archivo CSS desde la hoja de estilo LESS o SASS descargada.

Otra de las características más importantes que Bootstrap proporciona es que permite escalar, a cualquier dispositivo, sus aplicaciones de forma fácil y eficiente con una sola base de código.

#### 4.3.1.5. MySQL



MySQL es un sistema de administración y gestión de bases de datos relacionales.

Sus características fundamentales son: rendimiento, confiabilidad y facilidad de uso. Las cuales han convertido a MySQL en la principal opción de base de datos para aplicaciones Web.

Ha sido desarrollado bajo licencia dual GPL/Licencia comercial por Oracle Corporation.

Está desarrollado en su mayor parte en ANSI C y C++. Tradicionalmente, se considera uno de los cuatro componentes de la pila de desarrollo LAMP, WAMP y XAMPP.

MySQL permite crear base de datos y tablas, insertar datos, modificarlos, eliminarlos, ordenarlos, hacer consultas y realizar muchas operaciones, etc., resumiendo: administrar bases de datos.

MySQL es un interpretador de SQL (Structure Query Language -Lenguaje de Consulta Estructurado) que se trata de un lenguaje de programación para trabajar con base de datos relacionales.

### **4.3.2. Entorno empleado**

Toda la implementación de la herramienta se ha llevado a cabo en mi ordenador personal de forma local. Las dos herramientas que nos han facilitado dicho trabajo son: para el desarrollo del código y la organización de los archivos, el IDE NetBeans, y para el alojamiento web, el paquete XAMPP. Frente a una amplia variedad de herramientas con funcionalidades similares, ha resultado convincente escoger ambas gracias a la facilidad de uso que ofrecen y al soporte de los lenguajes de programación utilizados en este proyecto.

#### **4.3.2.1. NetBeans**

NetBeans es un entorno de desarrollo integrado libre, gratuito y de código abierto. Permite el uso de un amplio rango de tecnologías de desarrollo como son: Java, JavaScript, HTML5, PHP, C / C ++ y mucho más.

La plataforma NetBeans está escrita en el lenguaje de programación Java y permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

Entre sus características se encuentra un sistema de proyectos basado en Ant, control de versiones y refactoring, modularidad, buen editor de código, multilenguaje, con el habitual coloreado y sugerencias de código, acceso a clases pinchando en el código, control de versiones, localización de ubicación de la clase actual, comprobaciones sintácticas y semánticas, plantillas de código, coding tips, herramientas de refactorización... y un largo etcétera.

#### 4.3.2.2. XAMPP

Xampp es un paquete de instalación independiente de plataforma, software libre que ha sido principalmente diseñado para ser fácil de instalar y usar pues provee de una configuración totalmente funcional desde el momento que lo instalas.

Es una distribución de Apache que consiste en en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl.

El programa se distribuye bajo la licencia GNU y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris y Mac OS X.

Además, proporciona un módulo denominado phpMyAdmin, para llevar a cabo la administración de las bases de datos de MySQL a diferencia de utilizar consola.

### **4.3.3. Librería FusionCharts Suite XT**

Una forma de mostrar los datos estadísticos será mediante gráficas. Para ello, se ha recurrido a una librería que permite generar gráficas adaptadas a las necesidades del programador, es decir, permite personalizarlas con nuestros datos y darles el formato que se desea. Esta librería es conocida con el nombre de FusionCharts Suite XT.

FusionCharts es la biblioteca de gráficos JavaScript más completa, con más de 90 gráficos y 1000 mapas. Estos gráficos se muestran perfectamente en cualquier dispositivo PC, Mac, iPad, iPhones, Android y dispositivos de Windows. Está escrita en PHP.

Comprende desde gráficos básicos (línea, columna, pastel, etc) a los más complejos (cascada, gantt, candelabro...).

FusionCharts Suite XT está repleto de funciones de alimentación que le permiten tener un control total sobre la estética y el comportamiento de todos los gráficos. Además, la API de JavaScript de cada gráfico expone una mayor funcionalidad, como actualizaciones de datos del lado del cliente, actualización de parámetros individuales, exportación de gráficos como imágenes o PDF, creación de perfiles multinivel.

FusionCharts viene con complementos de código abierto para las bibliotecas populares (jQuery), marcos (AngularJS & React) y lenguajes (ASP.NET & PHP).

La elección de esta librería se debe principalmente a su capacidad por procesar datos que se obtienen de una base de datos y mostrarlos de manera inmediata, lo cual facilita la integración en nuestro proyecto. Además, presenta una extensa documentación y demos en tiempo real, lo que hace una implementación más sencilla.

### **4.3.4. Implementación de la herramienta**

Como se ha comentado sobre la lógica de negocio, distinguimos entre la capa de presentación y la base de datos. Por ello, el proyecto consta de dos directorios en los que se localizan cada uno de los ficheros que corresponden a estas dos partes. Todos estos ficheros son de tipo php, incluyendo los que pertenecen a la



Creación de una herramienta para la catalogación de empresas, orientada a estudios de gestión

capa de presentación, que, a pesar de estar escritos, mayoritariamente, con lenguaje html, tienen embebidos bloques de php. Estos bloques mediante las etiquetas: <?php y ?> permitirán al servidor web interpretar el código que hay entre ellas generando HTML y enviarlo al cliente como si fuera una página web estática, por eso es necesaria la extensión .php.

Además, se ha creado una carpeta para las hojas de estilo y el resto de carpetas que componen el proyecto son librerías necesarias para su funcionamiento. El árbol de directorios resultante es el siguiente:

#### **A. basedatos**

- búsqueda.php
- búsqueda\_tabla.php
- cargarSelect.php
- cargaranyos.php
- chartData.php
- conexion.php
- variaciones.php

#### **B. presentación**

- header.php
- footer.php
- index.php
- resultados.php

#### **C. css**

- estilo.css

#### **D. bootstrap**

#### **E. imágenes**

#### **F. fuentes**

## G. fusioncharts-suite-xt

A partir de esta estructura me basaré para explicar el código del producto detallando el funcionamiento y la utilidad de cada uno de estos archivos siguiendo un orden de relevancia.

### 4.3.4.1. Archivos php

Comenzaré por uno de los ficheros más importantes en nuestro proyecto, *conexión.php*.

- *conexion.php*

Este fichero es incluido en la mayoría de los *.php*, tanto aquellos dedicados a realizar operaciones con la base de datos, como en los responsables de la presentación. Su importancia se debe a ser el encargado de integrar *php* con *MySQL* estableciendo una conexión con nuestra base de datos denominada *TFG*.

La conexión a la API clásica de *MySQL* ha sido declarada obsoleta por *php*, por tanto, se ha utilizado otra API de integración conocida como *MYSQLi* para acceder a sus funcionalidades. Además, se hace uso de la función `set_charset('utf8')` que establece *utf8* como formato de codificación de caracteres predeterminado para enviar datos desde y hacia el servidor de la base de datos y evitar así problemas de acentos y caracteres especiales.

El código es el siguiente:

```
$servername = "localhost";  
$username = "usuario";  
$password = "usuario123";  
$dbname = "TFG";  
  
$conn = new mysqli($servername, $username,  
$password,$dbname);
```



```
$conn->set_charset("utf8");

// Check connection

if ($conn->connect_error) {

    die("Connection failed: " . $conn->connect_error);

}
```

Antes de entrar en detalle con el resto de archivos, cabe mencionar que han sido creados dos php independientes con la cabecera y el pie de página. De esta manera, se evita la sobrecarga y la repetición de código al tratarse de partes comunes en las vistas de la web. Bastará con referenciarlos en los archivos que corresponden con cada vista mediante las funciones `include('header.php')`, al inicio, y `include('footer.php')`, al final, dentro de bloques php.

- Header.php

Viene encabezado por la etiqueta `<html>` la cual permite indicar al navegador que lo que va a continuación debe ser interpretado como código html.

A continuación, le sigue la etiqueta `<head>` que define la cabecera del documento. En ella se indican los metadatos, el nombre que aparecerá en la barra de navegación y se vinculan las hojas de estilo necesarias para el correcto funcionamiento. En nuestro caso, se han incluido nuestro archivo de estilo (`estilo.css`), la hoja de estilo necesaria para poder visualizar el código bootstrap y una hoja de estilo que proporciona Google para utilizar como tipografía 'Open Sans' en nuestra web.

Hasta ahora, las líneas de código que corresponden a la descripción anterior forman parte de la información que necesita el documento, pero no son visualizadas directamente por el usuario. A partir de las siguientes, se cierra la cabecera y se abre la etiqueta `<body>`, donde será definido todo el contenido que se muestra en el navegador.

Antes de nada, se incluye el fichero explicado anteriormente para abrir la conexión con la base de datos mediante:

```
<?php include('basedatos/conexion.php'); ?>
```



De esta manera, cuando se vea la necesidad de acceder a la base de datos para mostrar cualquier contenido solo habrá que llamar a la conexión \$conn creada en él.

Por último, se define un contenedor que mostrará una imagen de portada sobre la que aparece el título de la web y un subtítulo.

- Footer.php

Este fichero simplemente contiene un contenedor en el que se muestra como pie de página una franja con el título de la web y mi identificación, nombre completo y correo de contacto.

Al tratarse de una aplicación web, es necesario un fichero principal que cree la página inicial que se lanza cuando cualquier usuario acceda a dicha aplicación. El encargado de ello es *index.php*.

- index.php

Es el archivo encargado de mostrar el contenido de la página inicial.

Al tratarse de un archivo de la capa de presentación, se incrustan los dos ficheros previamente descritos, al comienzo, la cabecera y al final, el pie de página.

Después de la inclusión de la cabecera, se encuentran unas líneas de código que muestra un contenedor dividido en tres secciones: servicio, funcionamiento y sobre. En cada una de ellas, se hace una breve descripción de pocas líneas con la intención de explicar el concepto de la herramienta, su funcionamiento y la motivación.

Su continuación, es el bloque de código más importante, el encargado de mostrar el buscador y a partir del cual se generará un resultado u otro.

Este bloque es un panel con un formulario. Un formulario web es un contenedor de controles mediante los cuales se recolecta información ingresada por los usuarios. Su uso se realiza mediante la etiqueta <form>



seguido de unos atributos que determinan cómo y dónde será procesada la información.

En nuestro caso, se ha usado de la siguiente manera:

```
<form action="resultados.php" method="post">
```

Siendo *resultados.php* el archivo que gestionará la información que se especifique en el formulario y post el método utilizado para empaquetar el formulario. De esta manera, dentro del formulario habrá un botón de tipo "submit" que una vez sea pulsado los datos serán inmediatamente enviados al archivo *resultados.php* el cual los gestionará y mostrará los resultados correspondientes.

Dentro del formulario, encontramos 3 controles:

- Un conjunto de botones de radio que corresponden a cada uno de los filtros: tamaño, condición jurídica, sector económico, edad, comunidad autónoma, provincia e ingresos. El usuario elegirá una única opción entre el conjunto cuyo valor se almacenará en una variable denominada *filtro*, que será utilizada posteriormente.
- Un desplegable que inicialmente se encuentra desactivado y al seleccionar un botón de radio, se activará cargando dinámicamente su contenido con las distintas modalidades del filtro seleccionado. Por ejemplo, si seleccionamos el filtro sector económico, el contenido del desplegable será: total, industria, comercio, construcción y otros servicios. El valor de este desplegable se almacenará en la variable *columnas* y será o bien null, que es el valor por defecto si no se selecciona nada, o el nombre de la modalidad seleccionada.
- Un selector de opciones múltiples que corresponde a los años de los que se quiere obtener la información. Al igual que ocurre con el desplegable, éste control en un principio está desactivado y al seleccionar una entre los distintos botones de radio se activa y su contenido es cargado de forma dinámica en función al filtro seleccionado, ya que cada filtro posee disponibilidad de datos en unos años concretos. Su valor será almacenado en una variable denominada *años* cuyo contenido puede ser null, en el caso de que no se seleccione ningún año, y el resultado a mostrar será de todos los años disponibles, o un array en el que se almacenan los años seleccionados.

Para cargar el contenido dinámico de estos dos controles, ha sido necesario el uso de JavaScript. Como se ha comentado anteriormente, se ha empleado de una de sus librerías conocida como JQuery la cual

nos permite realizar lo mismo que con JavaScript, pero de manera más optimizada ahorrando líneas de código.

Al tratarse de un bloque pequeño de código, no ha sido necesario crear un archivo .js externo si no que ha sido incluido en el php entre las etiquetas:

```
<script type="text/javascript"> </script>
```

Y el código resultante es el siguiente:

```
$(document).ready(function() {  
    $('.radio').change(function () {  
        $('input:radio[name=filtro]:checked').each(  
            function () {  
                $('.form-control').attr('disabled', false);  
                elegido=$(this).val();  
                $.post('basedatos/cargarSelect.php', {  
                    elegido: elegido}, function(data){  
                        $('#columnas').html(data);  
                    });  
                $.post('basedatos/cargaranyos.php' {  
                    elegido: elegido }, function(data){  
                        $('#años').html(data);  
                    });  
            });  
    });  
});
```



En él se especifica que, cada vez que se detecte un cambio en el conjunto de botones de radio, se pasa el valor del mismo, que corresponde con el nombre del filtro seleccionado, a los archivos *cargarSelect.php* y *cargaranyos.php*. Ambos establecen una conexión con la base de datos y realizan una consulta para obtener y listar las modalidades del filtro seleccionado y una consulta para obtener y listar los años en los que hay información para el filtro seleccionado, respectivamente.

Para entender mejor lo que realizan, se adjuntan sus líneas de código:

#### -cargarSelect.php

```
<?php

include('conexion.php');

$filtero= filter_input(INPUT_POST, 'elegido');

echo '<option selected disabled>Sin
seleccion</option>';

$sql="SELECT column_name FROM
information_schema.COLUMNS WHERE TABLE_NAME =
'$filtero'";

$result = $conn->query($sql);

while ($row = $result->fetch_assoc()){

    if($row["column_name"]!='Anyo'){

        echo '<option id="opcion" value="'.
        $row["column_name"].'">' . $row["column_name"].'</
        option>';

    }

}}
```

#### -cargaranyos.php

```
<?php

include('conexion.php');

$filtero= filter_input(INPUT_POST, "elegido");
```

```

$sql = "SELECT Anyo FROM ".$filtro." ORDER BY Anyo
desc";

$result = $conn->query($sql);

while ($row = $result->fetch_assoc()) {

    echo '<option name ="años[]"
value='.$row['Anyo'].'>'.$row['Anyo'].' </option>';

}

```

El resultado de ejecutar los métodos post pasándoles el valor del filtro, es un bloque html que será cargado en cada uno de los controles: el resultado de *cargarSelect.php* se cargará en *#columnas* que corresponde al desplegable, y el resultado de *cargaranyos.php* se cargará en *#años* que se corresponde con el selector de opciones múltiples.

Para finalizar con la descripción de *index.php*, es conveniente mencionar que no sería posible el funcionamiento y la visualización de esta página sin la inclusión de dos JavaScript externos:

```

<script src="bootstrap/js/bootstrap.min.js"></script>

<script src="//code.jquery.com/jquery-1.12.4.js"/>

```

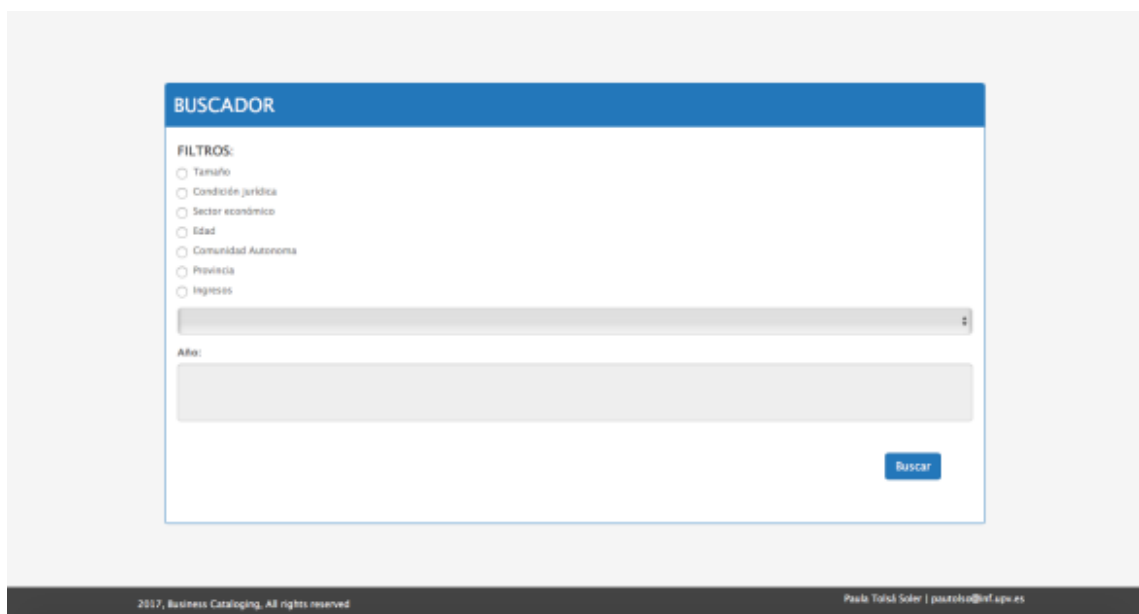
El primero, para poder comenzar a usar Bootstrap y el segundo, para el funcionamiento de los plugins de Bootstrap y para interpretar las funciones jQuery.

Una vez explicada la parte técnica de la página inicial, vamos a ver cuál es el resultado visual al ejecutarse en nuestro servidor web:



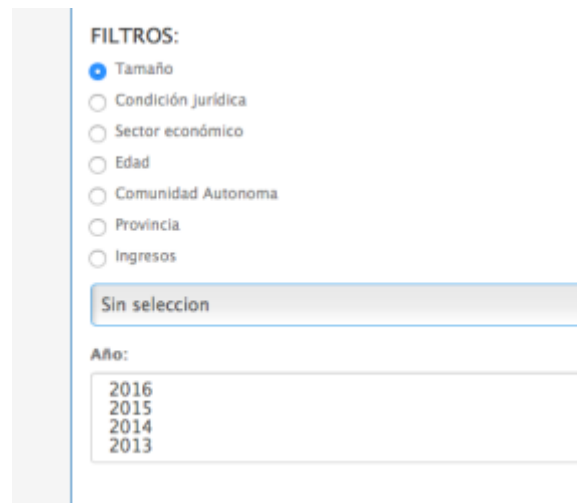


**Figura 9:** Cabecera y contenedor de la página inicial(index.php) (Fuente: Elaboración propia)



**Figura 10:** Panel de búsqueda con formulario y pie de página de página inicial(index.php) (Fuente: Elaboración propia)

En esta última captura, los controles del formulario del buscador referidos a la modalidad y a los años se encuentran desactivados y sin datos. Con el fin de visualizar el funcionamiento del buscador conforme se ha descrito anteriormente, se ha capturado un ejemplo:



**Figura 11:** Ejemplo de activación de los controles al pulsar el filtro ‘tamaño’  
(Fuente: Elaboración propia)



**Figura 12:** Ejemplo de carga del desplegable al seleccionar el filtro ‘tamaño’  
(Fuente: Elaboración propia)

Al seleccionar el filtro tamaño, el desplegable y el selector de opciones múltiples se ha activado y se han cargado los datos que le corresponden para poder continuar con la búsqueda.

Una vez explicada la página inicial junto con los archivos de la cabecera y pie de página, es hora de pasar al último archivo encargado de la vista que nos queda por exponer, *resultados.php*.

- resultados.php

Es el archivo que se ejecuta al accionar sobre el botón buscar de la página inicial y que gestiona toda la información del formulario.

Contiene un conjunto de etiquetas HTML que hacen posible una vista igual que la anterior, pero sustituyendo el contenido del buscador por un contenedor en el que se muestran todos los resultados de acuerdo a lo seleccionado previamente.

Para ello, se vuelven a incluir los archivos *header.php* y *footer.php* y un contenedor en el que se incorpora un elemento de navegación con tres pestañas que corresponden a las tres formas de presentar los resultados: tabla, gráficas y variaciones. Al ser un elemento facilitado por Bootstrap, al igual que en la página inicial, deberá ser incluido en el documento tanto el fichero *bootstrap.min.js* como la librería jQuery.

El uso de este elemento de navegación es el siguiente:

1. Se aplica la clase *.nav* para crear el elemento y después la clase *.nav-tabs* para mostrar los enlaces en forma de pestaña:

```
<ul class="nav nav-tabs">
  <li class="active"><a href="#tabla">Tabla</a></li>
  <li ><a href="#graficas">Gráficas</a></li>
  <li ><a href="#variaciones">Variaciones</a></li>
</ul>
```

2. Se crea una clase contenedora con *.tab-content* en la que aparecerá el contenido de una pestaña especificado en su *tab-pane* correspondiente.

```
<div class="tab-content">
  <div id="tabla" class="tab-pane fade in active ">
  </div>
```



```
<div id="graficas" class="tab-pane fade"></div>
<div id="variaciones" class="tab-pane fade"></div>
</div>
```

3. Es necesario unas líneas de código JavaScript para enlazar y mostrar cada pestaña con su contenido al ser pulsada.

```
$(".nav-tabs a").click(function(){
    $(this).tab('show');
});
```

Dentro de cada uno de los elementos con la clase `.tab-pane` se incluye el código que mostrará los resultados. A pesar de que cada una de las pestañas es una forma distinta de presentarlos, el procedimiento es el mismo en los tres casos:

1. Mediante la función `post` se recoge el valor de las variables del formulario:

```
-filtro: $filtro= filter_input(INPUT_POST, "filtro");
```

(que indica el valor del botón de radio del formulario seleccionado, es decir, el nombre del filtro)

```
-opción: $opcion= filter_input(INPUT_POST, "columnas");
```

(indica el valor del desplegable con el identificador `#columnas`, o nulo si no se ha seleccionado ninguno o el nombre de la modalidad seleccionada, que se corresponde con una de las columnas de la tabla del filtro elegido)

-años: hace referencia al valor del selector de opciones múltiples que indica los años. En el caso de que se haya seleccionado uno o más años, se crea una cadena de caracteres vacía en la que se van añadiendo los años seleccionados separados por comas y colocados entre paréntesis para su posterior uso en la consulta sql para extraer la información.

```
$años=' ';
```

```
if(isset($_POST['años'])) {  
    if (is_array($_POST['años'])) {  
        $años = "(";  
        foreach($_POST['años'] as $value){  
            if($años=="("){$años = $años."".$value;}  
            else{$años = $años.",".$value;}  
        }  
    }  
    $años = $años.")";  
}
```

En el otro caso, es decir, que no se haya seleccionado ningún año, la consulta se realizará sobre todos los años disponibles.

2. Se abre una conexión con la base de datos incluyendo el archivo *conexion.php* y a partir del valor de las tres variables obtenidas se hace una consulta en lenguaje SQL del tipo:

```
$sql1 = "SELECT * FROM ".$filtro." ORDER BY Anyo desc"
```

(si no se ha seleccionado ningún año)

```
$sql1 = "SELECT * FROM ".$filtro." WHERE Anyo IN  
".$años." ORDER BY Anyo desc"
```

(si se ha seleccionado uno o más años en concreto)

```
$result1 = mysqli_query($conn,$sql1);
```

3. `$result` es tratado de forma específica según la pestaña en la que nos encontremos.

Estos tres pasos junto con otras comprobaciones son los que se realizan en el archivo *búsqueda.php*, el cual es incluido en las tres secciones que le corresponden al contenido de cada pestaña mediante la etiqueta:

```
<?php include('busqueda.php');?>
```

Explicada la parte común a los tres contenedores, pasaremos a detallar sus contenidos en particular:

### **-Pestaña TABLA:**

```
<div id="tabla" class="tab-pane fade in active ">
    <div class="table-responsive">
        <table class="table table-sm table-hover table-
bordered">
            <?php
                include('basedatos/busqueda.php');
                include('basedatos/busqueda_tabla.php');?>
        </table>
    </div>
</div>
```

Como se puede observar, se incluye el archivo *busqueda\_tabla.php*, cuya función principal es generar el contenido de la tabla de forma dinámica según los resultados de la consulta que se realiza en el fichero *busqueda.php*, partiendo de los valores ingresados en el formulario de inicio.

Pueden ocurrir dos casos que se separan por una estructura de control if - else. O bien, que se haya seleccionado una modalidad, por tanto, se creará una tabla con dos columnas: la columna año y la columna asociada a la modalidad. O que ninguna modalidad haya sido escogida y, por tanto, habrá tantas columnas como modalidades tenga el filtro. Para obtener todas las modalidades disponibles del filtro se hace la siguiente consulta:

```
$sql2 = "SELECT column_name FROM
information_schema.COLUMNS WHERE
TABLE_NAME='$filtro'";

$result2 = mysqli_query($conn,$sql2);
```

(El resultado será un conjunto de filas con el nombre de cada columna de la tabla.)

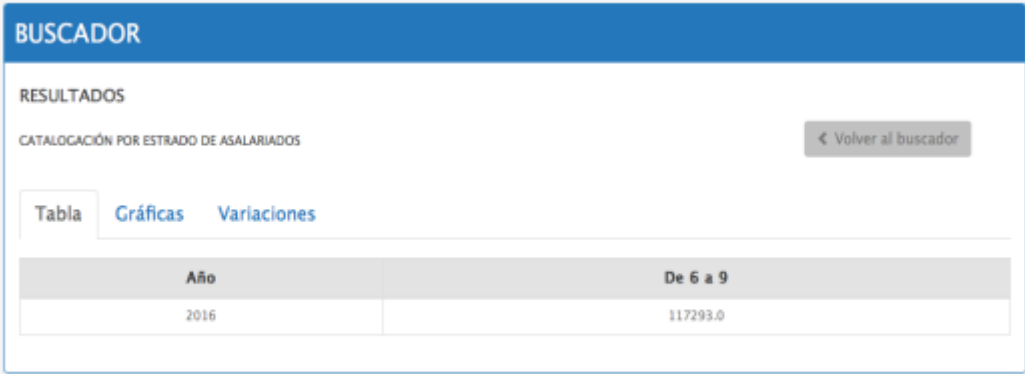


El código completo de *busqueda\_tabla.php* es:

```
$titulo_tabla='';
$array_nombres = array();
echo"<thead><tr>";
if($opcion==null){
    while($fila_nombre_col = $result2->fetch_assoc()) {
        echo"<th>".
            $fila_nombre_col["column_name"]."</th>";
        array_push($array_nombres,
            $fila_nombre_col["column_name"] );
    }
    echo"</tr></thead><tbody>";
    while($row1 = $result1->fetch_assoc()) {
        echo "<tr>";
        foreach ($array_nombres as $nombre ) {
            echo "<td>".$row1[$nombre]."</td> ";
        }
        echo "</tr>";
    }else{
        $titulo_tabla=$opcion;
        echo"<th>Año</th>";
        echo"<th>".$opcion."</th>";
        echo"</tr></thead><tbody>";
        if ($result1->num_rows > 0) {
            while($row = $result1->fetch_assoc()) {
                echo "<tr><td> ".$row['Anyo']."</td> ";
                echo "<td> ".$row[$opcion]."</td></tr> ";
            }
        }
    }
}
```

```
    }  
else {echo "0 results";}  
}  
echo"</tbody>";
```

Para conocer el resultado visual del archivo que se está describiendo (*resultados.php*), concretamente, de la pestaña tabla, se van a adjuntar algunas capturas:



Año	De 6 a 9
2016	117293.0

**Figura 13:** Ejemplo de la página resultados al seleccionar el filtro 'tamaño', la modalidad 'de 6 a 9' en el año 2016 (Fuente: Elaboración propia)

**BUSCADOR**

RESULTADOS

CATALOGACIÓN POR ESTRADO DE ASALARIADOS ← Volver al buscador

Tabla **Gráficas** Variaciones

Año	De 6 a 9
2016	117293.0
2015	112527.0
2014	110819.0
2013	125029.0
2012	131944.0
2011	138040.0
2010	143016.0
2009	151233.0
2008	160460.0
2007	158159.0
2006	154635.0
2005	145443.0
2004	143417.0
2003	136679.0
2002	131666.0
2001	126050.0

**Figura 14:** Ejemplo de la página resultados al seleccionar el filtro 'tamaño' y la modalidad 'de 6 a 9' (sin elegir años) (Fuente: Elaboración propia)

**BUSCADOR**

RESULTADOS

CATALOGACIÓN POR ESTRADO DE ASALARIADOS ← Volver al buscador

Tabla **Gráficas** Variaciones

Año	Total	Sin asalariados	De 1 a 2	De 3 a 5	De 6 a 9	De 10 a 19	De 20 a 49	De 50 a 99	De 100 a 199	De 200 a 499	De 500 a 999	De 1000 a 4999	De 5000 o ms
2016	3216182.0	1791909.0	895574.0	202403.0	117293.0	75022.0	40895.0	12020.0	6243.0	3489.0	951.0	674.0	108.0
2015	3188878.0	1734002.0	899802.0	287430.0	112527.0	71518.0	39101.0	11505.0	5928.0	3340.0	937.0	685.0	107.0
2014	3119310.0	1672483.0	921000.0	284612.0	110819.0	70236.0	38157.0	11223.0	5753.0	3346.0	942.0	646.0	101.0
2013	3146570.0	1681588.0	922646.0	280643.0	125029.0	74204.0	39506.0	11745.0	6130.0	3450.0	885.0	643.0	101.0
2012	3198117.0	1764987.0	867550.0	288896.0	131944.0	79113.0	43070.0	12700.0	6434.0	3214.0	935.0	689.0	105.0
2011	3250576.0	1795321.0	847912.0	313408.0	138040.0	84834.0	46160.0	13279.0	6585.0	3288.0	923.0	679.0	107.0
2010	3291263.0	1778005.0	895095.0	318155.0	143016.0	86390.0	48765.0	13745.0	7098.0	3374.0	952.0	651.0	101.0
2009	3355830.0	1767470.0	919092.0	332671.0	151233.0	101601.0	53641.0	15075.0	7672.0	3578.0	1006.0	678.0	111.0
2008	3422239.0	1754374.0	958711.0	345848.0	160460.0	110369.0	61709.0	16242.0	8061.0	4511.0	1117.0	725.0	112.0
2007	3338657.0	1706140.0	932526.0	340638.0	158159.0	108066.0	60738.0	13737.0	7780.0	4218.0	1072.0	684.0	99.0
2006	3174393.0	1616883.0	881748.0	328820.0	154635.0	105470.0	58725.0	15087.0	7267.0	3908.0	999.0	658.0	98.0
2005	3064129.0	1574166.0	850367.0	310787.0	145443.0	100211.0	56260.0	14394.0	7009.0	3817.0	967.0	630.0	84.0
2004	2942583.0	1500306.0	818149.0	303783.0	143417.0	98245.0	53267.0	13470.0	6650.0	3590.0	945.0	582.0	79.0
2003	2813159.0	1439918.0	769515.0	295611.0	138679.0	93823.0	51619.0	13486.0	6462.0	3540.0	878.0	530.0	78.0
2002	2710400.0	1425332.0	718914.0	271764.0	133066.0	89840.0	50595.0	13468.0	6190.0	3276.0	815.0	466.0	74.0
2001	2645317.0	1408792.0	692964.0	259764.0	126050.0	86308.0	48205.0	12953.0	5867.0	3180.0	744.0	418.0	72.0

**Figura 15:** Ejemplo de la página resultados al seleccionar el filtro 'tamaño' (sin elegir ni modalidad ni años) (Fuente: Elaboración propia)

## -Pestaña GRÁFICA:

Si el usuario desea visualizar los datos resultantes en forma de gráfica, esta es la pestaña que deberá de pulsar para que se muestren en el contenedor de contenidos.

El primer paso a la hora de programar esta sección ha sido incluir la librería fusioncharts-suite-xt para poder comenzar a hacer uso de las gráficas. Pero no solo basta con añadir dicho archivo php, sino que ha sido necesario incluir varios ficheros JavaScript al documento para conseguir su pleno funcionamiento:

```
<script src="fusioncharts-suite-xt/js/fusioncharts.js"/>
```

```
<script src="fusioncharts-suite-xt/js/themes/fusioncharts.theme.fint.js"/>(Para el tema de los gráficos)
```

Se ofrece al usuario la posibilidad de elegir entre cuatro tipos distintos de gráficas: diagrama de columnas, diagrama de pastel, de área y donut. Independientemente del tipo que se elija todas ellas representan la información en función de los años de acuerdo a una modalidad en concreto del filtro elegido. Por ello, es necesario que se haya seleccionado una modalidad. En el caso de que no sea así, antes de mostrar las gráficas se cargará un desplegable con las distintas opciones a elegir.

El bloque de código que corresponde al contenedor de contenido de la pestaña gráficas se muestra a continuación.

```
<div id="graficas" class="tab-pane fade">
  <?php include(`fusioncharts-suite-xt/
fusioncharts.php`);?>
  <div class="graficas">
    <div hidden="true" id="opcion_desplegable1">
      <select id="opcion_graficas" class="form-control">
        <?php
          include 'basedatos/busqueda.php';
```

## Creación de una herramienta para la catalogación de empresas, orientada a estudios de gestión

```
        echo '<option selected disabled>Seleccione una
opcion</option>';

        while($row2 = $result2->fetch_assoc()) {
            if($row2['column_name'] != 'Anyo') {
                echo '<option value="'
$row2['column_name'].'">'.$row2['column_name'].'</
option>';}}?>
        </select>
    </div>
<div class="row" id="radio-buttons">
    <label class="radio-inline">
        <input checked="checked" type="radio"
        name="tipo_graficas" value="column" >Column chart
    </label>
    <label class="radio-inline">
        <input type="radio" name="tipo_graficas"
        value="pie" >Pie chart
    </label>
    <label class="radio-inline">
        <input type="radio" name="tipo_graficas"
        value="area" >Area chart
    </label>
    <label class="radio-inline">
        <input type="radio" name="tipo_graficas"
        value="doughnut" >Doughnut chart</label>
```





```

</div>
    <div id="nombre_grafica" >
        <?php echo $titulo_tabla;?>
    </div>
    <div id="grafica" ></div>
</div>
</div>

```

En él se crea el desplegable que carga las modalidades del filtro que ha elegido el usuario. Esto se realiza incluyendo el archivo *busqueda.php* para establecer la conexión con la base de datos y realizar la consulta con la variable filtro introducida en el formulario que recoge el nombre de todas las columnas de dicha tabla. El resultado es un conjunto de filas que son recorridas e insertadas en el desplegable.

También se crean los cuatro botones que hacen referencia a los tipos de gráfica a mostrar.

Por último, le siguen dos etiquetas `<div>`, una para cargar el título de la gráfica, es decir, la modalidad, y otro en el que se cargará la gráfica en sí.

El bloque anterior no es suficiente por sí mismo para lograr la función que tiene dicha pestaña. Por este motivo, son necesarias unas líneas de código JavaScript que serán las responsables de dar el comportamiento adecuado a la página y crear las gráficas cargando de forma dinámica sus datos.

Estas líneas de código serán ejecutadas una vez se cargue en su totalidad la página web(DOM) gracias a formar parte de la función jQuery comprendida entre las etiquetas html `<script></script>`:

```
$(document).ready(function(){});
```

Debido a la importancia de ellas, a continuación, se van a detallar paso por paso, distinguiendo entre los dos casos posibles: que haya una modalidad elegida o no.

El valor de la modalidad se obtiene mediante:



```
var opcion= '<?php echo $opcion;?>';
```

Y a partir de él se realiza una comprobación con una estructura de control if-else para conocer si ha sido elegida o no. En el caso de que sí, la variable *opción* será distinto de una cadena vacía y en el caso contrario, su valor será una cadena vacía.

#### A. Con modalidad seleccionada(opcion!="):

En este caso, una vez sea seleccionada la pestaña GRAFICAS, inmediatamente se cargará en el div con identificador "nombre\_grafica" la opción seleccionada como título de la gráfica y en el div con identificador "grafica" la gráfica con los datos correspondientes.

El encargado de que esto ocurra es el siguiente bloque de código:

```
var filtro= '<?php echo $filtro;?>';
var opcion= '<?php echo $opcion;?>';
var string_años= '<?php echo $años;?>';

$.post('basedatos/chartData.php',
{filtro:filtro,opcion:opcion,string_años:string_años}
, function(data){
    var dataChart= JSON.parse(data);
    var chartProperties = {
        paletteColors: '#0075c2,#1aaf5d,#f2c500',
        labelDisplay: 'auto',
        baseFontColor: '#333333',
        baseFont: 'Helvetica Neue,Arial',
        captionFontSize: '14',
        subcaptionFontSize: '14',
        subcaptionFontBold: '0',
        showBorder: '0',
```

```

        bgColor: '#ffffff',
        showShadow: '0',
        canvasBgColor: '#ffffff',
        showCanvasBorder: '0',
        showHoverEffect: '1',
        theme: 'fint'
    };

    var apiChart = new FusionCharts({
        type: 'column2d',
        renderAt: 'grafica',
        width: '600',
        height: '350',
        dataFormat: 'json',
        dataSource: {
            "chart": chartProperties,
            "data": dataChart
        }
    });

    apiChart.render();

```

Como se puede observar, se ha creado un objeto FusionCharts indicándole todos los parámetros que necesita, así como: estilo y propiedades del gráfico, el tipo de gráfico, el elemento del documento donde va a ser renderizado y los datos a mostrar.

Al tratarse de un objeto que va a ser creado cada vez a partir de unos datos distintos, no pueden ser introducidos como constantes. De esta manera, los datos del gráfico serán recogidos de una variable denominada dataChart cuyo valor es una cadena en formato json formada por pares (label,value) con los que se rellenará.

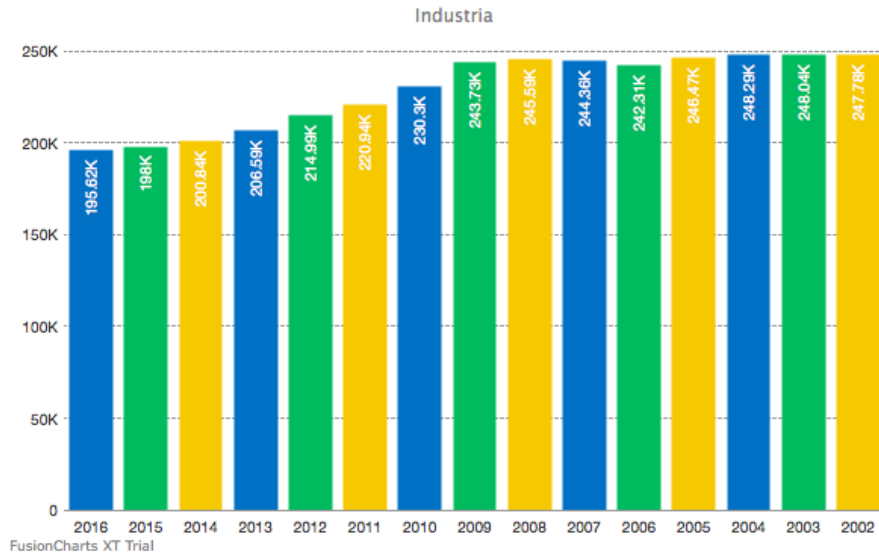
El contenido de dicha variable es el resultado de ejecutar el archivo chartData.php. Este archivo abre una conexión con la base de datos y realiza una consulta sobre ella para obtener todos los datos existentes de la columna correspondiente a la modalidad seleccionada de la tabla elegida como filtro en los años escogidos, si es que los hay, en el caso de lo haber, se realizará sobre todos los años disponibles. Pero para que las variables modalidad, filtro y años tengan valor en este archivo, ha sido necesario hacer uso de la función \$.post) sobre él pasándole dichos valores, como se puede ver en el código anterior.

Así pues, una vez realizada la consulta, se ha creado un array sobre el que se han ido añadiendo cada una las filas del resultado con el fin de generar el json mencionado el cual es interpretado al crear el objeto FusionCharts.

Como aclaración se adjunta parte del chartData.php:

```
$jsonArray = array();  
  
    // Añade los datos al array  
while($row1 = $result1->fetch_assoc()) {  
    $jsonArrayItem = array();  
    $jsonArrayItem['label'] =$row1["Anyo"];  
    $jsonArrayItem['value'] =$row1[$opcion];  
    array_push($jsonArray, $jsonArrayItem);  
}  
  
$jsonEncodedData = json_encode($jsonArray);  
echo $jsonEncodedData;
```

Con el fin de materializar lo explicado hasta el momento sobre la sección GRAFICAS en el caso de que una modalidad haya sido seleccionada, he realizado una búsqueda sobre las empresas según el sector económico de la industria desde el año 2002 al 2016 y el resultado visual ha sido el siguiente:



**Figura 16:** Ejemplo de los datos en el sector económico de la industria en forma de gráfica de columnas) (Fuente: Elaboración propia)

Con respecto a los datos, el resultado del archivo chartData.php ha sido:

```
[{"label":"2016","value":"195619.0"}, {"label":"2015","value":"198004.0"}, {"label":"2014","value":"200835.0"}, {"label":"2013","value":"206585.0"}, {"label":"2012","value":"214992.0"}, {"label":"2011","value":"220935.0"}, {"label":"2010","value":"230301.0"}, {"label":"2009","value":"243729.0"}, {"label":"2008","value":"245588.0"}, {"label":"2007","value":"244359.0"}, {"label":"2006","value":"242310.0"}, {"label":"2005","value":"246472.0"}, {"label":"2004","value":"248289.0"}, {"label":"2003","value":"248038.0"}, {"label":"2002","value":"247779.0"}]
```

**B. Con modalidad sin seleccionar(opcion==""):**

En este otro caso, una vez sea seleccionada la pestaña GRAFICAS, se activará un desplegable pero el contenido no será mostrado hasta que se detecte que una de sus opciones haya sido seleccionada. Con ello, el div con identificador "grafica" cargará la gráfica con los datos adecuados según la modalidad seleccionada.

1. El primer paso, será mostrar el desplegable cuyo contenido ha sido creado anteriormente con todas las posibles opciones del filtro de interés. Para esto se hace visible el div que lo contiene cuyo identificador es *#opcion\_desplegable1*:

```
$("#opcion_desplegable1").show();
```

2. Posteriormente, se crea una función que es llamada cada vez que el desplegable cambia de valor con el fin de obtener el valor de la opción seleccionada:

```
$('#opcion_graficas').change(function(event) {  
  
    opcion2=$('#opcion_graficas').val();
```

3. Al igual que en el caso anterior, se hace uso de la función post para pasar al archivo *chartData.php* los valores del filtro, modalidad y años y obtener así el json con todos los datos a mostrar.

Pero la diferencia es que, en este caso, al no haber seleccionado inicialmente una modalidad, el valor de esta variable será el valor de la opción del desplegable elegida, es decir:

```
$.post('basedatos/chartData.php' {  
  
    filtro:filtro,  
  
    opcion:opcion2,  
  
    string_años:string_años}, function(data) {  
  
        var dataChart= JSON.parse(data);
```

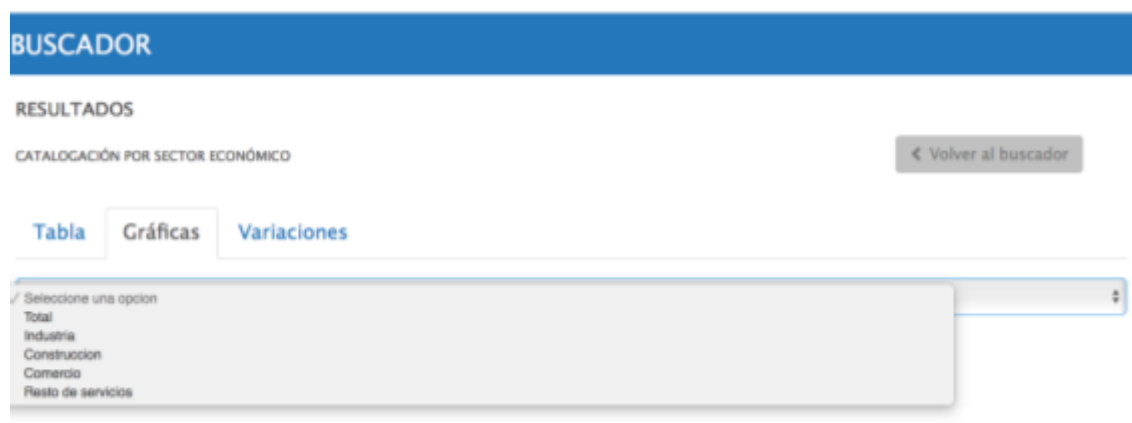
4. La forma en la que se crea la gráfica es la misma que la descrita anteriormente y se coloca dentro de la función post la cual pertenece a la función explicada en el punto 2. Esto es debido a que el documento

se encuentra a la escucha de un cambio en el desplegable en todo momento, pues con este cambio el contenido de la gráfica cambia dinámicamente.

Esta alteración al seleccionar una nueva modalidad es debida a que se crea una gráfica nueva cada vez sobre la anterior.

Para visualizar este caso, se ha realizado una búsqueda sobre la población empresarial según su sector económico comprendida entre los años 2002 y 2016.

El primer resultado que se ha obtenido al cargar la página de los resultados y seleccionar la pestaña GRAFICAS es el siguiente:



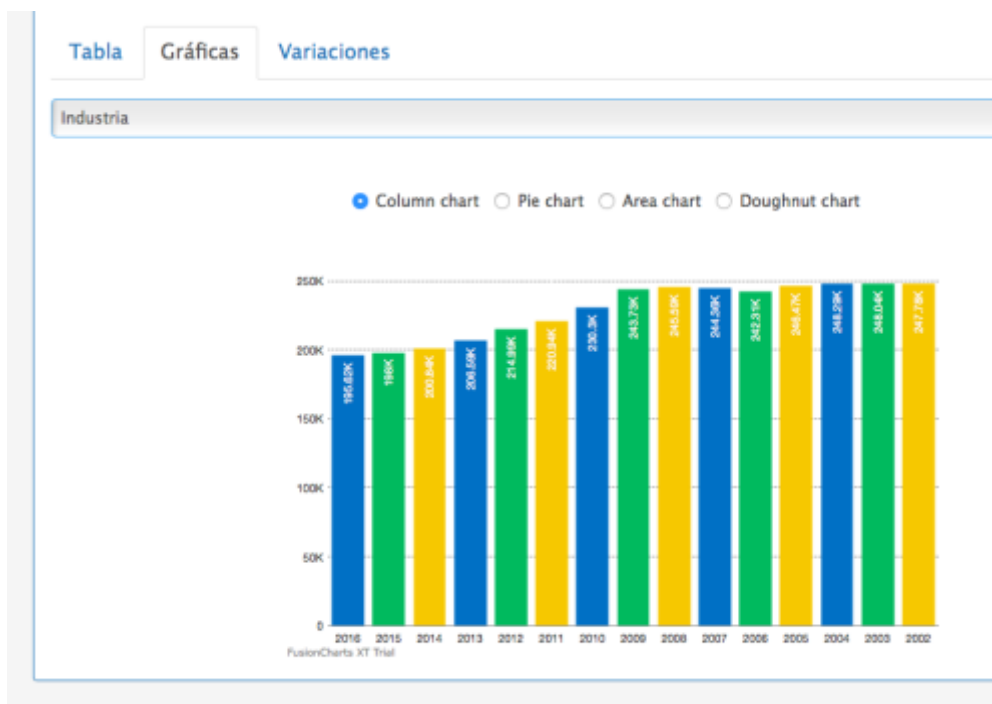
**Figura 17:** Pestaña GRAFICAS al haber realizado una búsqueda sin selección de modalidad) (Fuente: Elaboración propia)

Al seleccionar una de las opciones, por ejemplo, el sector del comercio, se observa que el archivo *chartData.php* se ha ejecutado correctamente generando los datos con los que se ha creado la gráfica:



**Figura 18:** Pestaña GRAFICAS al haber seleccionado posteriormente la modalidad) (Fuente: Elaboración propia)

Si se cambia de opción se puede observar que se genera un nuevo archivo chartData.php con los datos correspondientes a partir del cual se crea una nueva gráfica:



**Figura 19:** Resultado al cambiar de modalidad (Fuente: Elaboración propia)



Para finalizar con la sección encargada de mostrar los resultados en forma de gráficas y partiendo de lo mencionado sobre la posibilidad de elegir entre varios tipos de gráficas, se va a explicar la función que permite mostrar la misma gráfica en distintos tipos.

Esta función es `chartType (chartType, options)`, es ofrecida por la propia librería `fusionCharts` y se aplica sobre el objeto `FusionChart` creado clonándolo según el tipo indicado como parámetro y renderizándolo de nuevo.

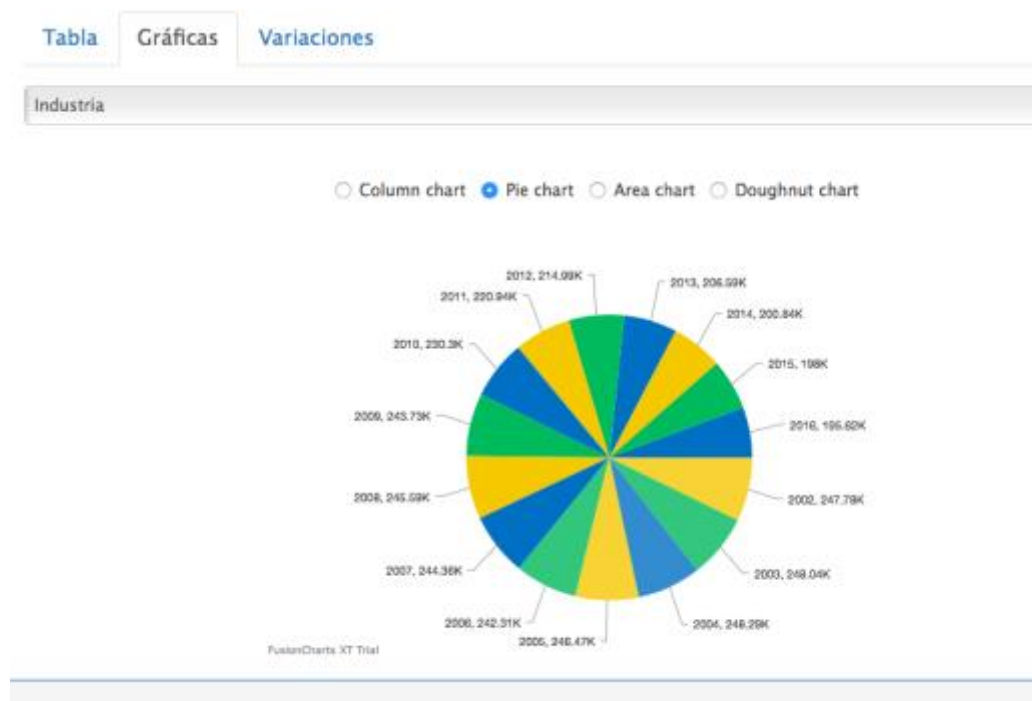
La forma en la que se elige el tipo que se desea es mediante botones de radio. Como se quiere que la gráfica cambie dinámicamente cada vez que se acciona sobre uno de estos botones, se necesitará llamar a la función `chartType()` cada vez que haya un cambio en los botones. Por tanto, se utilizará la función `jQuery`:

```
$( 'input:radio[name="tipo_graficas"]' ).on( 'change',  
function(event) {  
    switch(this.value){  
        case 'pie':  
            apiChart.chartType('pie2d');  
            break;  
        case 'area':  
            apiChart.chartType('area2d');  
            break;  
        case 'doughnut':  
            apiChart.chartType('doughnut2d');  
            break;  
        default:  
            apiChart.chartType('column2d');  
            break;  
    }  
}
```



El uso del switch es para obtener el valor del botón seleccionado y así se pueda pasar a determinar el tipo de gráfica que se desea.

Para observar el correcto funcionamiento de estos botones y de las llamadas a la función encargada del cambio de tipo de gráficas, se ha adjuntado un ejemplo de una gráfica en forma de pastel con los datos empresariales sobre el sector económico de la industria en todos los años disponibles.



**Figura 20:** Ejemplo de gráfica 'pie chart' (Fuente: Elaboración propia)

### -Pestaña VARIACIONES

Esta última sección corresponde con una funcionalidad adicional que se le ha querido dar a la herramienta.

Se trata de realizar comparaciones de los resultados con un cierto año base. De esta manera, se podrá saber cuál es la diferencia de la cantidad de empresas entre un año y el año base de cualquiera de las modalidades de un filtro. Su fin es concluir si este balance es positivo o negativo, es decir, si ha habido un aumento o una pérdida en el número de empresas.

Al igual que con las gráficas, nos encontramos con el problema de que la tabla que se va a mostrar en este apartado es en función de los años de acuerdo a una modalidad en concreto, por tanto, si no se selecciona

ninguna, antes de mostrar la tabla de las variaciones, un desplegable con las distintas opciones será cargado para que se seleccione una de ellas y así que los resultados sean cargados.

Además de este desplegable y debido a que es necesario indicar el año base sobre el que se desea obtener las variaciones, se deberá crear otro desplegable que cargue los posibles años entre los que seleccionar la base.

Estos dos elementos junto con sus correspondientes etiquetas explicativas y un elemento <div> sobre el que se cargará la tabla son creados en el contenedor de contenidos de las variaciones que pertenece al archivo que se está describiendo(*resultados.php*) y es su código es el siguiente:

```
<div id="variaciones" class="tab-pane fade">
  <div id="data" ></div>
  <div hidden="true" id="opcion_desplegable2">
<label id="alerta">
  <label style="color: red">*</label>
  Para poder continuar debe de seleccionar una de las
  siguientes opciones
</label>
<select class="form-control" id="opcion" >
  <?php include 'basedatos/busqueda.php';
  echo '<option selected disabled> Seleccione una
opcion</option>';
  while($row2 = $result2->fetch_assoc()) {
    if($row2['column_name']!= 'Anyo') {
      echo '<option value="' . $row2['column_name'] . '">' .
$row2['column_name'] . '</option>';
    }
  }
  }?>
```



```
</select>
</div><br><br>
<label>Año base:</label>
<div class="form-group">
    <select class="form-control" id="año_base">
        <?php include 'basedatos/busqueda.php';
        echo '<option selected disabled>Seleccione año
base</option>';
        while($row1 = $result1->fetch_assoc()) {
            echo '<option value='.\$row1['Anyo']. '>'.
\$row1['Anyo']. '</option>';
        }?>
    </select>
<label id="etiqueta">
<label style="color: red">*</label>
Seleccione un año base para obtener las variaciones
con respecto a dicho año
</label>
<div id="tabla_variaciones"></div>
</div>
</div>
```

Pero para darle la funcionalidad adecuada a esta sección se necesitará añadir varias líneas de código en la función jQuery:

```
$(document).ready(function() {})
```

En primer lugar, se deberá comprobar si se ha seleccionado o no una modalidad, mediante un if-else que compare si el valor de la variable que corresponde con la modalidad es una cadena vacía con contiene valor.

Si se ha elegido una modalidad en el formulario inicial, al accionar esta pestaña se cargará directamente el desplegable con todos los años entre los que seleccionar la base.

En el caso de que no se haya elegido, se mostrará el desplegable en el que se cargan todas las opciones dejando bloqueado al que lista los años base hasta el momento en el que se detecte una selección en el primero. Todo esto queda determinado:

```
var opcion= '<?php echo $opcion;?>';  
  
if(opcion==''){  
    $("#opcion_desplegable2").show();  
    $('#etiqueta').hide();  
    $('#año_base').attr('disabled', true);  
    $('#opcion').change(function(event) {  
        $('#año_base').attr('disabled', false);  
        $('#alerta').hide();  
        opcion= $('#opcion').val();  
        $('#data').html(opcion);  
    })  
}
```

Una vez haya sido seleccionado el año base, en la etiqueta div con identificador #tabla\_variaciones se cargará, por un lado, un cuadro de texto con el dato del año base y, por el otro, una tabla cuyas filas son los distintos años sobre los que se va a realizar la comparación y sus columnas son: año, nº de empresas, diferencia y diferencia en porcentaje.

El contenido de este elemento variará en función del año base seleccionado sin necesidad de recargar la página.

Esto se debe a una función jQuery que atiende los cambios en el desplegable, de tal manera que al elegir un año, este se almacena en una variable que, posteriormente, junto con las variables: título, filtro, opción y array\_años(un array con todos los años con los que se compare el año base) son enviados mediante la función post a un archivo denominado *variaciones.php*.

El código correspondiente con lo descrito es el siguiente:

```
$('#año_base').change(function(event) {  
    var año_base= $('#año_base').val();  
    var array_años = [];  
    $("#año_base option").each(function () {  
        array_años.push($(this).text());  
    });  
    $('#etiqueta').hide();  
    $.post('basedatos/variaciones.php', {  
        año_base:año_base,  
        array_años:array_años,  
        titulo:titulo,  
        filtro:filtro,  
        opcion:opcion}, function(data){  
        $('#tabla_variaciones').html(data);  
    });  
});
```

El archivo *variaciones.php* es el encargado de realizar una consulta a la base de datos para obtener los datos necesarios y crear una tabla con el contenido y cálculos adecuados.

Debido a su importante labor que desempeña en esta sección, se va a explicar parte por parte:

1. Se obtienen los valores que se le pasan por medio del método post:

```
$año_base= filter_input(INPUT_POST, "año_base");
```

```

$titulo= filter_input(INPUT_POST, "titulo");
$filtero= filter_input(INPUT_POST, "filtro");
$array_años= filter_input(INPUT_POST, "array_años");
$opcion=filter_input(INPUT_POST, "opcion");

```

**2. Se realiza la consulta para obtener los datos del año base en la modalidad seleccionada:**

```

$datos_año_base = "SELECT `".$opcion.`` FROM
".$filtro." WHERE Anyo =".$año_base."";

$resultado_base = $conn-> query($datos_año_base) or
trigger_error($conn -> error." ".$datos_año_base);

$fila_año_base = mysqli_fetch_assoc($resultado_base);

```

**3. Se crea la tabla, sus columnas y se le da valor al cuatro de texto que muestra el dato del año base a comparar:**

```

echo "<div class='table-responsive'>
<table class='table table-sm table-hover table-
bordered'>
<div class='div1'>
<h3>Número de empresas en el año $año_base
$fila_año_base[$opcion]</h3></div>
<thead>
<tr>
<th>Año</th>
<th>Empresas</th>
<th>Diferencia con respecto al $año_base</th>
<th>Diferencia %</th>";

```



```
echo"</tr></thead><tbody>";
```

4. Se recorre el array de los años realizando una consulta por cada año y mostrando los datos como una nueva fila de la tabla. Además, con cada dato se realizan dos sencillos cálculos:

-para obtener la diferencia, se resta el dato del año base menos el dato de cada año.

-para obtener la diferencia en %, se divide el dato del año base entre cada uno de los datos de los años, al resultado se le resta 1 y se multiplica por 100.

El resultado de estos cálculos puede ser positivo o negativo, y para que resulte más representativo se le da color al texto en función de si el número ha crecido o ha disminuido.

```
foreach($_POST['array_años'] as $value){
    if($value!=$año_base && $value!= 'Seleccione año
base'){
        $datos_año_comparar="SELECT `".$opcion.``
FROM `.`
$filtro." WHERE Anyo =".$value;
$resultado_comparar = mysqli_query($conn,
$datos_año_comparar);
$fila_año_comparar =
mysqli_fetch_assoc($resultado_comparar);
$diferencia=$fila_año_base[$opcion]-
$fila_año_comparar[$opcion];
$porcentaje =round((( $fila_año_base[$opcion]/
$fila_año_comparar[$opcion])-1)*100,2);
echo "<tr>
```



```

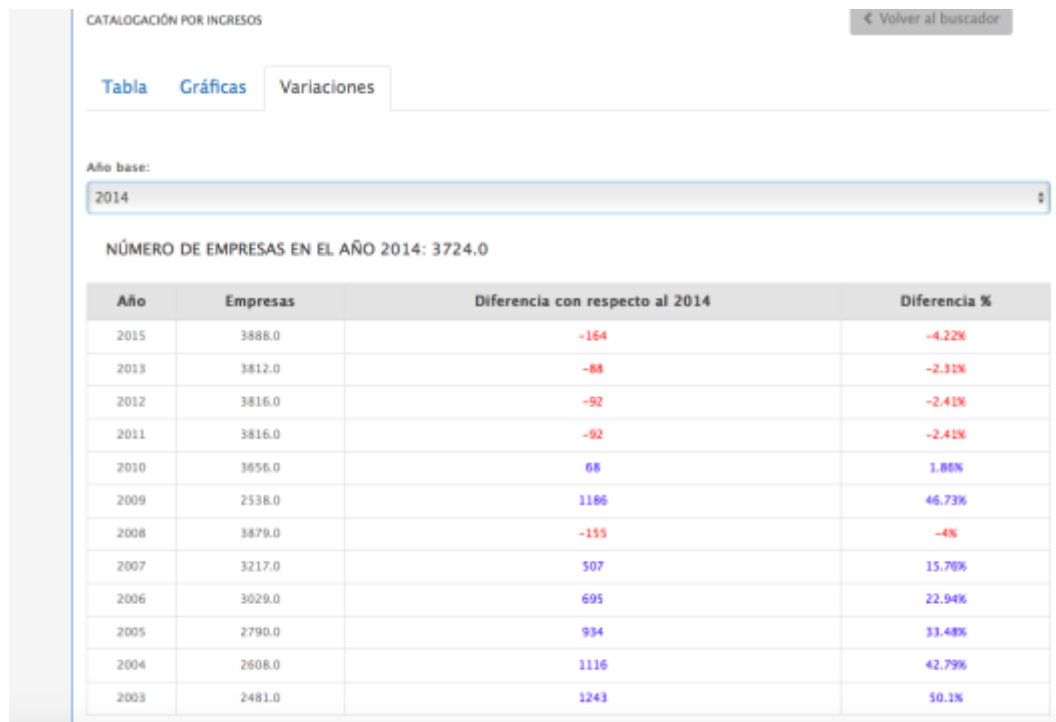
<td>$value</td>
<td>$fila_año_comparar[$opcion]</td>";
    if($diferencia>=0){
        echo "<td id='diferencia' style='color:
        blue'>$diferencia</td>";
        echo "<td id='diferencia' style='color:
        blue'>$porcentaje%</td>";
    }else {
        echo "<td id='diferencia' style='color:
        red'>$diferencia</td>";
        echo "<td id='diferencia' style='color:
        red'>$porcentaje%</td>"; }
    echo "</tr>";
}}
}

```

Como paso final, se demostrará el funcionamiento de todo este código escrito anteriormente mediante una serie de capturas de la página resultados al escoger el apartado variaciones:



**Figura 21:** Ejemplo de la pestaña VARIACIONES al buscar número de empresas por ingresos  $\geq 50$  millones de euros (Fuente: Elaboración propia)



**Figura 22:** Variaciones del número de empresas por ingresos  $\geq 50$  millones de euros con respecto al año 2014 (Fuente: Elaboración propia)



**Figura 23:** Ejemplo de la pestaña VARIACIONES al buscar por ingresos sin elegir ninguna modalidad (Fuente: Elaboración propia)

10-49 Millones euros

10-49 Millones euros

Año base:

2015

NÚMERO DE EMPRESAS EN EL AÑO 2015: 13972.0

Año	Empresas	Diferencia con respecto al 2015	Diferencia %
2014	13339.0	633	4.75%
2013	13869.0	103	0.74%
2012	14418.0	-446	-3.09%
2011	14787.0	-815	-5.51%
2010	14916.0	-944	-6.33%
2009	23881.0	-9909	-41.49%
2008	16708.0	-2736	-16.38%
2007	13933.0	39	0.28%
2006	12949.0	1023	7.9%
2005	11979.0	1993	16.64%
2004	11269.0	2703	23.99%
2003	10561.0	3411	32.3%

**Figura 24:** Variaciones del número de empresas por ingresos entre 10 y 49 millones de euros con respecto al año 2015 (Fuente: Elaboración propia)

#### 4.3.5. Problemas en la implementación

A pesar de que finalmente se logrado conseguir el desarrollo por completo, no han sido poco los fallos que han ido apareciendo durante la implementación. Como es evidente, no se van a describir dichos problemas, aunque he considerado oportuno mencionar algunos que me han parecido interesantes. Uno de los que más ha dado que hablar ha estado relacionado con el cotejamiento de la base de datos. Como cada una de las tablas no ha sido creada desde cero si no que se ha creado a partir de un archivo .csv, dicha importación no permite realizarse en formato utf-8 si no en ISO 8859-1. Utf-8 es la codificación de caracteres que permite a los navegadores interpretar el lenguaje en español, incluyendo las 'ñ' y los acentos. Por ello, tanto al hacer consultas sobre la base de datos como al mostrar sus resultados los caracteres especiales no eran reconocidos como tal, sino que se representaban mediante símbolos y letras no correspondidas lo cual conllevaba a resultados inesperados. Tras el uso de varias funciones de conversión cada vez que se hacía una consulta o se iban a mostrar los resultados, finalmente se encontró una solución global al problema siguiendo los pasos:

1. Introducir en la cabecera de los códigos html la siguiente línea:

```
<meta http-equiv="Content-Type" content="text/html;  
charset=UTF-8" />
```

2. Establecer el conjunto de caracteres predeterminado a usar cuando se envían datos desde y hacia el servidor de la base de datos sobre la conexión abierta en el archivo `conexion.php`:

```
$conn->set_charset("utf8");
```

3. Cambiar el tipo de codificación desde el `phpmyadmin` tanto de la base de datos, como cada una de las tablas y sus respectivas columnas a `latin1_spanish_ci`.

En varias ocasiones, durante la ejecución de la web para probar el desarrollo, el contenido no era cargado y se desconocía el motivo de ello al estar aparentemente todo programado correctamente. Finalmente se conoció que la causa era el orden de inclusión de los archivos `JavaScript` necesarios en la página. Es necesario que el fichero `jquery.js` se encuentre por delante de todos para que el resto entren en funcionamiento ya que estos dependen de él. A continuación, deben incluirse el resto de dependencias como, por ejemplo: `bootstrap` y la librería `fusionCharts` y por último el código `JavaScript` escrito al final de la página.

## 5. Pruebas

---

Una vez logrado el funcionamiento de nuestro sistema, se observará el resultado al ejecutarlo en unas circunstancias previamente especificadas con el fin de realizar una evaluación sobre ciertos aspectos: corrección, robustez, eficiencia, usabilidad..., es decir, se evaluará la calidad del software desarrollado.

Para ello se han realizado un conjunto de actividades que nos ayudaran a concluir la calidad del proyecto y la adecuación con los requisitos iniciales.

### 5.1. Validación CSS

Se ha hecho uso de una hoja de estilo denominada `estilo.css` para el diseño de la página web. Con el fin de validarla se ha utilizado el validador de css ofrecido por el consorcio W3C dando como resultado “CSS Versión 3 válido” sin ningún error encontrado.

#### Resultados del Validador CSS del W3C para `estilo.css` (CSS versión 3)

**¡Enhorabuena! No error encontrado.**

¡Este documento es [CSS versión 3](#) válido!

**Figura 25:** Variaciones del número de empresas por ingresos entre 10 y 49 millones de euros con respecto al año 2015 (Fuente: [https://jigsaw.w3.org/css-validator/#validate\\_by\\_upload](https://jigsaw.w3.org/css-validator/#validate_by_upload))

## 5.2. Compatibilidad con navegadores

Desde el principio del desarrollo se ha utilizado como navegador predeterminado Safari. Una vez realizada la implementación se ha considerado necesario probar su funcionamiento en otros navegadores como: Google Chrome, Internet Explorer y Firefox.

No ha habido ningún cambio destacable con respecto a Safari. La aplicación web mantiene su diseño, funcionamiento y velocidad de carga. La única diferencia casi inapreciable, es el cambio de la tipografía, pues a pesar de que se incluye una fuente de Google Fonts en los documentos, cada navegador prioriza sus ajustes de texto y tamaño.

Por tanto, se concluye que la aplicación es compatible con cualquier navegador, hecho que le otorga una mayor accesibilidad.

## 5.3. Análisis resolución de pantalla

La página web también ha sido utilizada y testeada con distintas resoluciones gracias a Google Chrome que ofrece entre las herramientas del desarrollador un simulador de dispositivos móviles que permite modificar la resolución de la pantalla que se está simulando de forma manual.

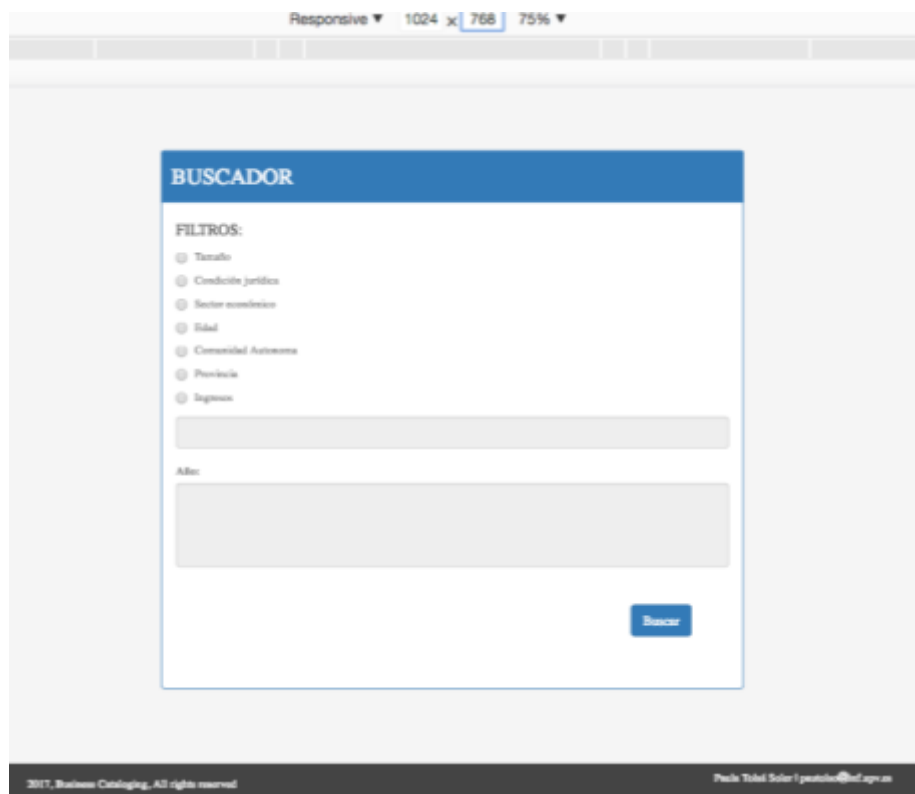
El tamaño de cualquier pantalla no acostumbra a superar las 10 pulgadas y una resolución de 1024x768 ppp. Además de esta resolución, se han probado otras de las más utilizadas por los dispositivos de sobremesa en la actualidad.

Gracias a la utilización del framework Bootstrap, se ha creado una aplicación web “responsive”, es decir, que se ajusta a cualquier tamaño de pantalla que use el usuario sin tener que indicarlo mediante líneas de código. Por tanto, el resultado ha sido satisfactorio en todas ellas:

<b>1024 x 768</b>	<b>OK</b>
<b>1920 x 1080</b>	<b>OK</b>
<b>1280 x 1024</b>	<b>OK</b>
<b>1366 x 768</b>	<b>OK</b>

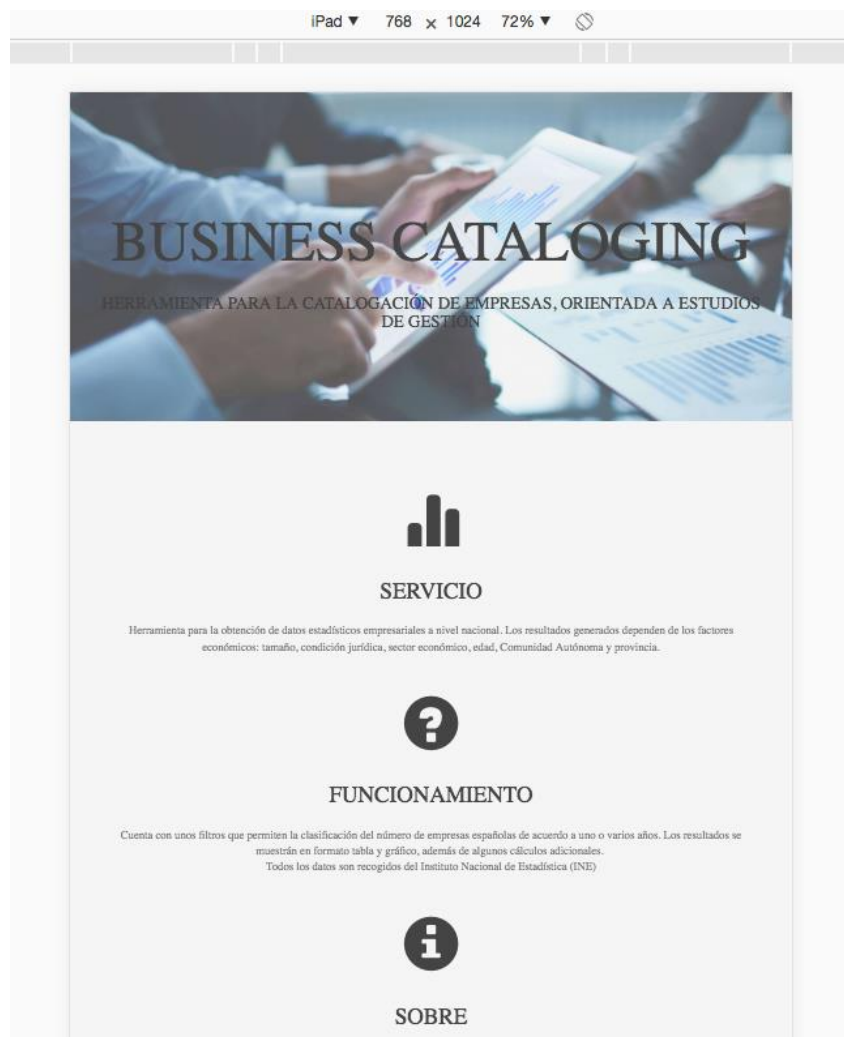


**Figura 26:** Visualización de la pantalla inicial con una resolución de 1024X768 (Fuente: Elaboración propia)



**Figura 27:** Visualización de la pantalla inicial con una resolución de 1024X768 (Fuente: Elaboración propia)

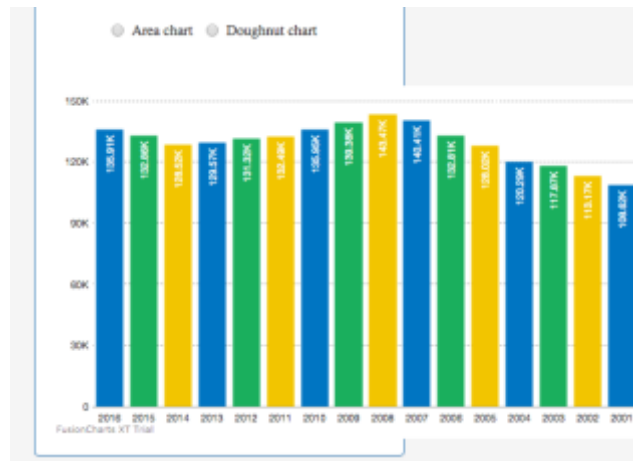
A pesar de que nuestra herramienta ha sido pensada para ser utilizada desde un computador, se han probado resoluciones que corresponden con pantallas más pequeñas como puedan ser las de dispositivos móviles y tablets. Se ha observado que a partir de una resolución horizontal inferior a 1000, cambia de diseño ofreciendo una apariencia móvil como se muestra a continuación.



**Figura 28:** Visualización de la pantalla inicial con una resolución de 758X1024 (iPad) (Fuente: Elaboración propia)

Sin embargo, a medida que vamos reduciendo las resoluciones, el aspecto se desestructura. Vemos que todos los elementos intentan adaptarse, exceptuando las gráficas que al tener un tamaño fijo no se reajustan, sobresaliendo de su contenedor. También ocurre con el formulario de la página inicial impidiendo la utilización de la herramienta.





**Figura 29:** Gráfica sobresalida de su contenedor con una resolución de un dispositivo móvil (Fuente: Elaboración propia)

iPhone 5 ▾ 320 x 568 100% ▾

## BUSCADOR

**FILTROS:**

- Tamaño
- Condición jurídica
- Sector económico
- Edad
- Comunidad Autónoma
- Provincia
- Ingresos

Año:

**Figura 30:** Vista del formulario con una resolución de 320x568 (Fuente: Elaboración propia)

## 5.4. Velocidad de carga

Al tratarse de una web con una única funcionalidad y con pocos archivos externos que cargar, la velocidad de carga deberá ser pequeña. Para comprobarlo se utiliza una opción en las herramientas de desarrollador del navegador Google Chrome que lo indica:

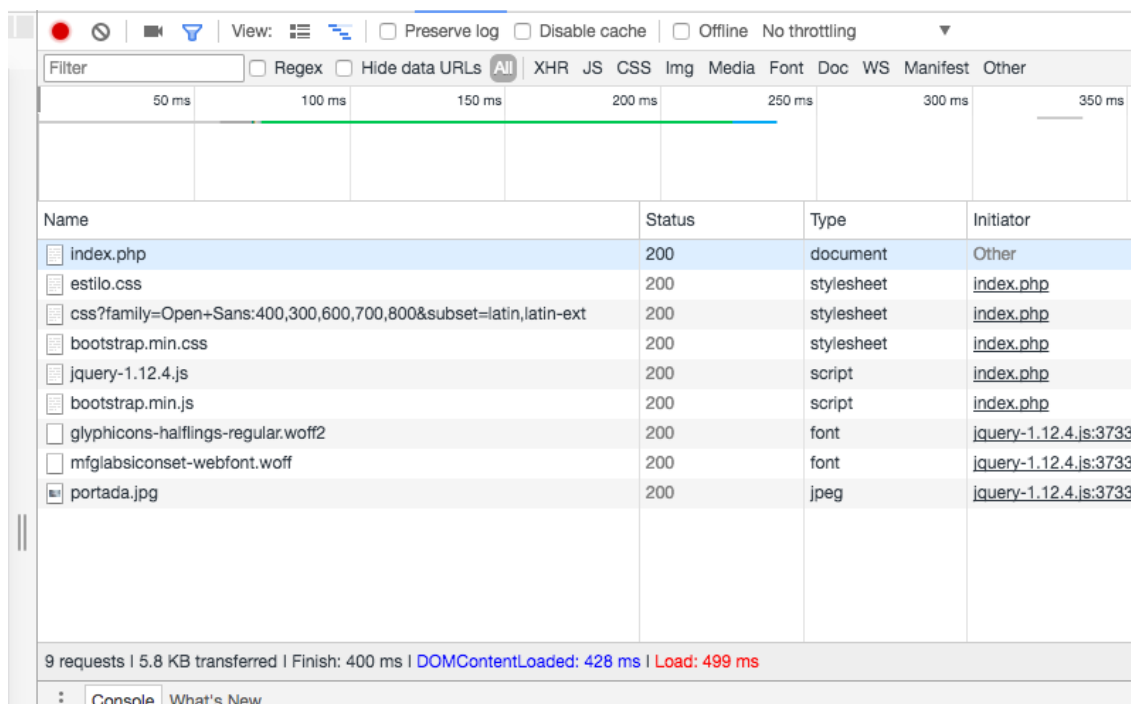


Figura 31: Velocidad al cargar la página principal (Fuente: Google Chrome)

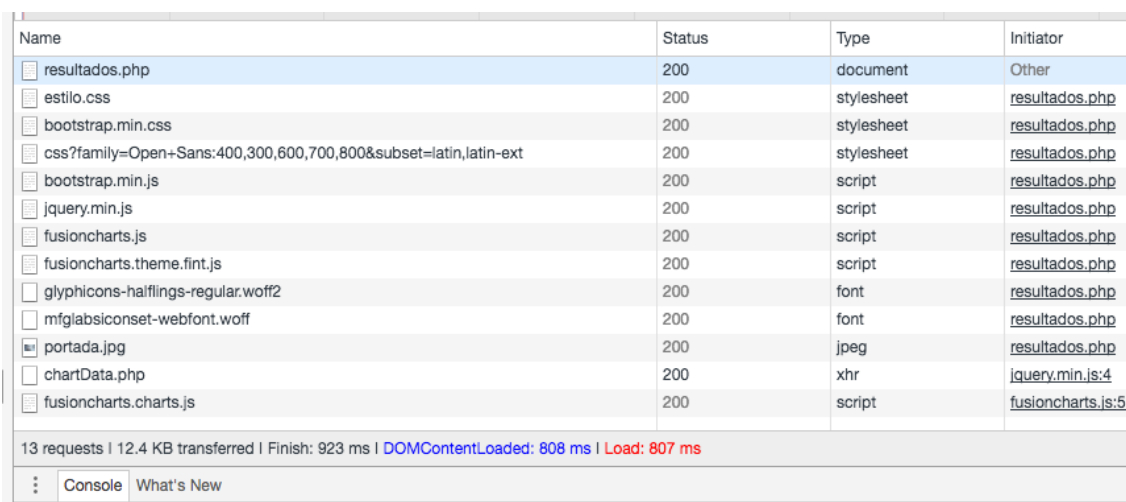


Figura 32: Velocidad al realizar una búsqueda (Fuente: Google Chrome)

En efecto, la velocidad de carga no llega al segundo, por lo que se puede concluir que es muy rápida.

## 5.5. Correspondencia de datos

Recapitulando a los objetivos de la herramienta, nos encontrábamos ante la necesidad de mostrar al usuario datos verídicos catalogados según distintas variables. Con este fin, se han realizado búsquedas paralelas con nuestra aplicación para ver si los resultados obtenidos coinciden.

Se ha monitorizado un ejemplo de estas búsquedas. Concretamente, la búsqueda de la población empresarial que tienen como condición jurídica sociedad anónima.

	Sociedades anónimas		
	2016	2015	2014
<b>Total</b>	85.430	89.392	92.986
<b>Sin asalariados</b>	27.400	28.151	27.662
<b>De 1 a 2</b>	15.086	16.201	17.433
<b>De 3 a 5</b>	8.868	9.777	10.555
<b>De 6 a 9</b>	7.860	8.356	9.091
<b>De 10 a 19</b>	9.356	9.847	10.679
<b>De 20 a 49</b>	8.859	9.065	9.435
<b>De 50 a 99</b>	3.669	3.683	3.766
<b>De 100 a 199</b>	2.101	2.094	2.122
<b>De 200 a 499</b>	1.395	1.394	1.417
<b>De 500 a 999</b>	424	419	431
<b>De 1000 a 4999</b>	344	335	324
<b>De 5000 o más asalariados</b>	68	70	71

**Figura 33:** Resultados por número de asalariados y condición jurídica sociedad anónima en los años 2016,2015 y 2014. (Fuente: DIRCE)

Año	Sociedades anónimas
2016	85430.0
2015	89392.0
2014	92986.0

**Figura 34:** Resultados de sociedades anónimas en los años 2016,2015 y 2014. (Fuente: Business Cataloging)

Como se comentó, en el apartado de comparación con la competencia, el DIRCE, únicamente muestra los datos en función de dos o más variables de catalogación lo cual justifica la primera de las imágenes anteriores. Si observamos en la fila del total de empresas por número de asalariados, los valores coinciden con los extraídos en nuestra propia herramienta. Así pues, se corrobora que los datos mostrados es información real.

## 5.6. Pruebas de uso

En esta actividad se ha probado el funcionamiento de la herramienta. Se ha sometido a todos los posibles hilos de ejecución:

PRUEBA	RESULTADO ESPERADO	RESULTADO OBTENIDO
Búsqueda catalogada por un filtro en concreto sin especificar modalidad ni año	<p>-Pestaña tabla: tabla con todas las modalidades como columnas y todos los años como filas.</p> <p>-Pestaña de gráficas: desplegable para seleccionar la modalidad, una vez seleccionada se cargan las gráficas correspondientes.</p> <p>-Pestaña variaciones: un desplegable para elegir la modalidad, una vez elegida se carga un desplegable del año base. Al elegir el año base, la tabla ya es mostrada. Si se cambia a un año base distinto, la información que se muestra también cambia.</p>	Resultado esperado
Búsqueda catalogada por un filtro en una modalidad concreta sin seleccionar años.	<p>-Pestaña tabla: tabla con dos columnas (año y la modalidad elegida) y con tantas filas como años disponibles hay información.</p> <p>-Pestaña gráficas: directamente se cargará una gráfica de columnas que después podrá ser modificado su tipo.</p> <p>-Pestaña variaciones: un desplegable para seleccionar el año base, una vez elegido se muestra la tabla con la</p>	Resultado esperado

	información adecuada y al cambiar de año base, los datos de la tabla se modifican.	
Búsqueda catalogada por un filtro en unos años determinados sin seleccionar la modalidad	<p>-Pestaña tabla: tabla con todas las modalidades disponibles del filtro como columnas y filas como tantos años hayan sido seleccionados.</p> <p>-Pestaña gráficas: desplegable para seleccionar la modalidad, una vez seleccionada se cargan las gráficas correspondientes.</p> <p>-Pestaña variaciones: un desplegable para elegir la modalidad, una vez elegida se carga un desplegable del año base. Al elegir el año base, la tabla ya es mostrada. Si se cambia a un año base distinto, la información que se muestra también cambia.</p>	Resultado esperado
Búsqueda catalogada por un filtro con una modalidad y unos años especificados.	<p>-Pestaña tabla: tabla con dos columnas (año y la modalidad elegida) y filas como tantos años hayan sido seleccionados.</p> <p>-Pestaña gráficas: directamente se cargará una gráfica de columnas que después podrá ser modificado su tipo.</p> <p>-Pestaña variaciones: un desplegable para seleccionar el año base, una vez elegido se muestra la tabla con la información adecuada y al cambiar de año base, los datos de la tabla se modifican.</p>	Resultado esperado

A partir de esta tabla se llega a la conclusión de que el funcionamiento de nuestro sistema es correcto.

Durante el proceso de ejecución de los distintos casos de prueba, no se han detectado a penas discrepancias con lo que se esperaba que se iba a obtener.

Únicamente, nos hemos encontrado con un error de implementación.

Este error corresponde con la creación de las gráficas. Concretamente, al no

seleccionar inicialmente una modalidad y seleccionarla desde el desplegable de la sección de las gráficas. En este caso, se crea una gráfica de columnas con los datos correspondientes independientemente del tipo de gráfica que esté pulsado en ese momento.

Ha sido corregido añadiendo en las líneas de código JavaScript una estructura switch que se ejecuta cada vez que un botón de radio cambia su valor. En esta estructura se actualiza una variable typeChart con el tipo de objeto FusionChart que será creado, pasándosela en dicha creación.

```
var typeChart='column2d';
$('input:radio[name="tipo_graficas"]').on('change', function(event) {
    switch(this.value){
    case 'pie':
        typeChart='pie2d';
        break;
    case 'area':
        typeChart='area2d';
        break;
    case 'doughnut':
        typeChart='doughnut2d';
        break;
    default:
        typeChart='column2d';
        break;
    }
});

var apiChart = new FusionCharts({
    type: typeChart,
    ...
});
```

Por lo demás, todo queda disponible para su uso y puesta en práctica.

## 6. Conclusiones

---

Es el momento de analizar los resultados obtenidos. Para ello, hay que mirar atrás y recapitular hacia los objetivos del proyecto.

El proyecto, como bien indica su titulación, tenía como objetivo principal llevar a cabo una herramienta que ofreciera a los usuarios la posibilidad de catalogar datos estadísticos empresariales del territorio español.

El resultado es una aplicación web reproduce lo descrito, por tanto, se puedo afirmar que he cumplido con la propuesta inicial del trabajo.

La parte más decisiva del proyecto ha sido la elección del tipo de herramienta a desarrollar. Hubiera sido muy sencillo partir de una herramienta ya existente y trabajar únicamente con la manipulación de los datos. Sin embargo, debido al punto en el que me encuentro de la carrera, he creído conveniente arriesgarme con una implementación de mayor envergadura, una web dinámica mediante la combinación de distintos lenguajes de programación.

Esta decisión ha sido todo un reto personal debido a que partía de escasos conocimientos en los lenguajes de programación web. Pero gracias a ella, he ampliado mis competencias en tecnologías utilizadas en el día a día que me serán de gran utilidad en un futuro.

Además, debido a mi desconocimiento en este ámbito, mi capacidad de autoaprendizaje ha aumentado, al igual que también me ha servido para aprender muchos detalles de programación, que desconocía hasta el momento, a partir de errores que me han ido surgiendo.

Por todo ello, puedo decir que ha supuesto para mí un crecimiento tanto profesional como personal hacia el mundo laboral que me espera.

Pero no solo ha sido aprender aspectos nuevos, sino que también me ha servido para poner en práctica los conocimientos adquiridos en el grado: aplicar el ciclo de vida del desarrollo software en un proyecto real, utilizar bases de datos y consultas en lenguaje MySql, diseño de interfaces y utilización de elementos de IGU... pero, lo más importante, a programar con pensamiento de ingeniero.

Y es que, me he dado cuenta de que cualquier persona con conocimientos básicos podría desarrollar un software. Sin embargo, va más allá de los



conocimientos, pues implica disciplinas como Ingeniería del software, gestión de proyectos, ergonomía, gestión de calidad, metodologías...

Por lo que respecta a las tecnologías empleadas, tengo que destacar la importante función que ha desempeñado el lenguaje JavaScript en mi proyecto, ya que, a pesar de no haberse utilizado con exceso, ha resultado crucial para dar dinamismo a nuestra aplicación y crear una sensación de comunicación entre el sitio web y el usuario. También, es destacable el ahorro de tiempo en el diseño web gracias al uso del framework de Bootstrap.

Como resumen general de esta conclusión, la elaboración del proyecto ha sido una experiencia positiva en todos los aspectos. Solo cabe decir que todo esfuerzo y dedicación tienen su recompensa, y así ha sido, se ha conseguido la finalidad de la que partíamos de forma satisfactoria, hecho que ha conllevado a una gran ampliación de mis conocimientos.

## **6.1. Posibles mejoras**

A pesar de que el proyecto ha sido concluido, siempre quedan algunas facetas que mejoraría.

Debido al desconocimiento del que partía desde un inicio de las tecnologías empleadas, aunque el desarrollo se ha realizado de la mejor manera posible, hay trozos de código que pueden ser optimizados al igual que se podría aplicar la refactorización para lograr un mayor entendimiento del código para un futuro mantenimiento.

Por lo que a funcionalidades nos referimos, implementaría alguna más que me hubiera gustado incluir, pero por motivos de tiempo no he llegado. Añadiría más filtros de catalogación. Implementaría otra pestaña adicional en la que se realizaran cálculos económicos sobre los resultados obtenidos para dar mayores facilidades a los estudios de gestión hacia los que se orientan las búsquedas. También permitiría que se pudieran hacer búsquedas sobre los movimientos de la población empresarial, es decir, altas, bajas y permanencias de las empresas catalogadas por los filtros que nos ofrece la herramienta. E incluso ofrecería la posibilidad de exportar los datos a otros tipos de documentos como pdf, excel, csv ...

En cuanto al diseño, podría ser mejorado para poderse adaptar a pantallas de dispositivos móviles.



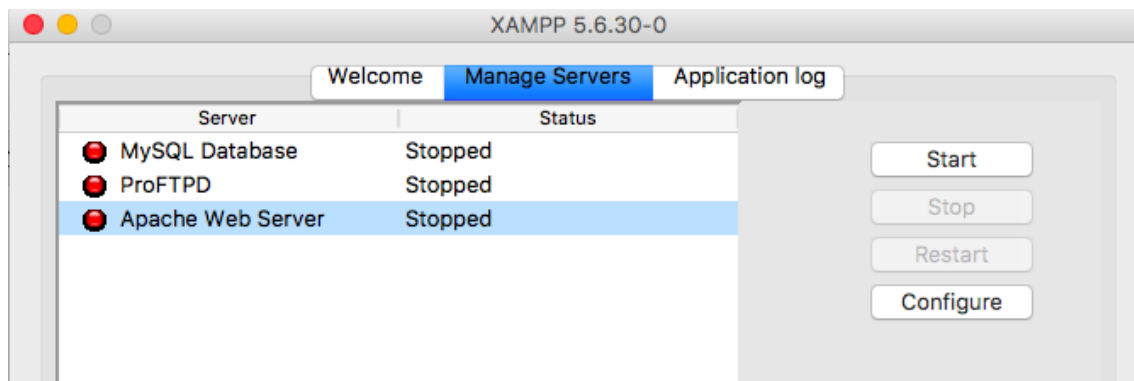
# 7. Anexo

---

## 7.1. Preparación del entorno

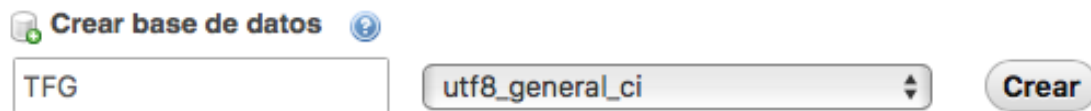
Para poder comenzar con el desarrollo de nuestra aplicación web, era necesaria la instalación de un sistema de gestión de bases de datos MySQL, el servidor web Apache y el intérprete para el lenguaje PHP. Estas tecnologías forman parte del paquete de instalación XAMPP, por lo que ha sido instalado mediante los siguientes pasos:

1. Descargar el paquete XAMPP de acuerdo a nuestro sistema operativo desde el enlace: <https://www.apachefriends.org/es/index.html>
2. Descomprimir el archivo descargado, ejecutarlo y seguir el asistente de instalación.
3. Seleccionar el lugar destino, donde se va a instalar y donde residirán los proyectos creados.
4. Una vez finalizada la instalación, abrimos el panel de control de XAMPP y activamos tanto Apache como MySql con el botón start.



**Figura 35:** Panel de control de Xampp con los servicios desactivados (Fuente: Xampp)

5. Se comprobará su funcionamiento accediendo a: <http://localhost/xampp/>  
Una vez instalado y funcionando correctamente, crearemos y configuraremos la base de datos. Se deberá de acceder a la URL <http://localhost/phpmyadmin/>  
Accionamos sobre el botón nueva base de datos y rellenaremos lo que nos pide:



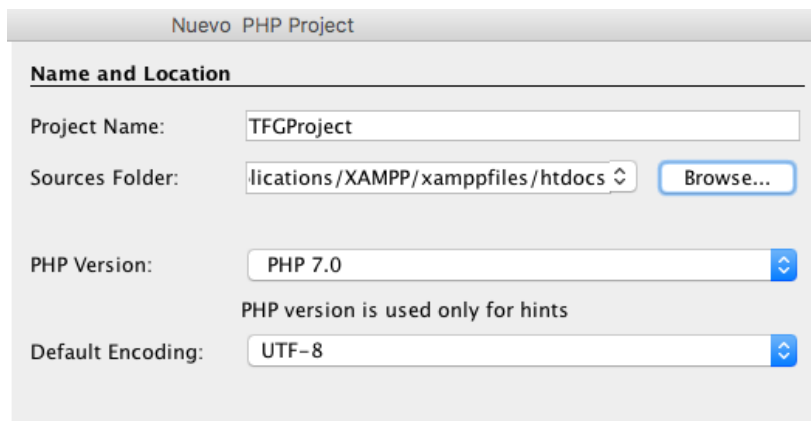
**Figura 36:** Nueva base de datos (Fuente: phpmyAdmin)

Una vez creada, está lista para añadir las tablas. Pero, antes de nada, se debe de crear un usuario con los permisos para acceder a ella desde php.

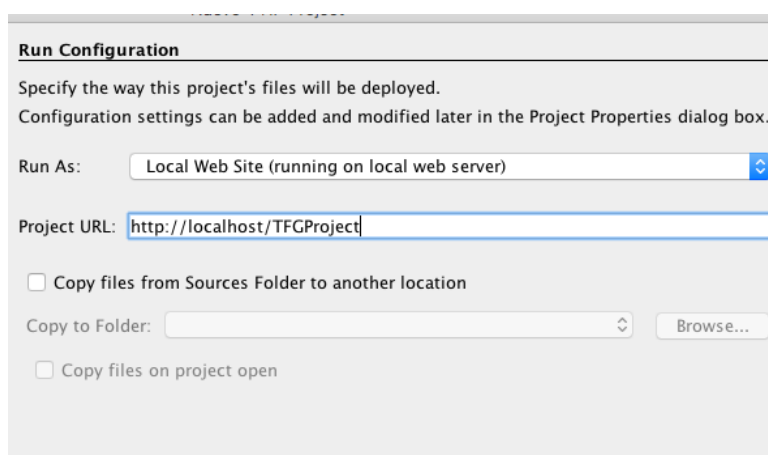
### Agregar cuenta de usuario

**Figura 37:** Nueva cuenta de usuario (Fuente: phpmyAdmin)

Con respecto al entorno de desarrollo, en nuestro caso NetBeans, se creará un nuevo proyecto PHP que será situado dentro de la carpeta XAMPP concretamente en una denominada htdocs.



**Figura 38:** Creación de un nuevo proyecto PHP (Fuente: NetBeans)



**Figura 39:** Configuración del nuevo proyecto (Fuente: NetBeans)

Con todos estos pasos, ya se podrá acceder a la aplicación con el enlace: <http://localhost/TFGProject>, que ejecutará el archivo `index.php` que modificaremos para mostrar nuestra página de inicio.

## 7.2. Importación de la base de datos

Como se ha comentado, las tablas de la base de datos no han sido creadas desde cero, si no que han sido creadas mediante la importación de archivos `.csv` extraídos de páginas web como es el DIRCE. Para la comprensión de este procedimiento, se va a mostrar un ejemplo paso por paso de cómo ha sido creada y poblada la tabla `ccaa` (Comunidades Autónomas).

En el DIRCE, los datos de las Comunidades Autónomas no están todos agrupados. Existe una tabla de Empresas por CCAA, actividad principal (grupos CNAE 2009) y estrato de asalariados con los datos de nuestro interés del 2016

Creación de una herramienta para la catalogación de empresas, orientada a estudios de gestión

al 2008 y otra de Empresas por CCAA, actividad principal (grupos CNAE93) y estrato de asalariados con los datos 2009 al 1999.

Como los datos que queremos están en función de otras variables: actividad principal y estrato de asalariados, habrá que indicar que queremos obtener los resultados de todas las comunidades autónomas con respecto al total de las otras dos variables, es decir:

- En la tabla de Empresas por CCAA, actividad principal (grupos CNAE 2009) y estrato de asalariados

**Empresas por CCAA, actividad principal (grupos CNAE 2009) y estrato de asalariados.**  
Unidades: Empresas

- Seleccione valores a consultar

Comunidades y Ciudades Autónomas	Estrato de asalariados	Actividad principal	Periodo
<input type="text"/> Nacional Andalucía Aragón Asturias, Principado de Balears, Illes Canarias Cantabria Seleccionados: 20 Total: 20	<input type="text"/> Total Sin asalariados De 1 a 2 De 3 a 5 De 6 a 9 De 10 a 19 De 20 a 49 Seleccionados: 1 Total: 13	<input type="text"/> Total CNAE 05 Extracción de antracita, hulla y lignito 051 Extracción de antracita y hulla 052 Extracción de lignito 06 Extracción de crudo de petróleo y gas natural 061 Extracción de crudo de petróleo 062 Extracción de gas natural Seleccionados: 1 Total: 334	<input type="text"/> 2016 2015 2014 2013 2012 2011 2010 Seleccionados: 9 Total: 9

- Elija forma de presentación de la tabla

	Comunidades y Ciudades Autónomas		
	Actividad principal		
	Estrato de asalariados		
Periodo	-	-	-

Total: 20 series y 180 celdas  
Consultar selección Consultar todo

**Figura 40:** Filtros para la obtención de los datos por CCAA del 2016 al 2008 (Fuente: INEBase-DIRCE)

- En la tabla Empresas por CCAA, actividad principal (grupos CNAE93) y estrato de asalariados se realizará una búsqueda del mismo tipo, pero para obtener los resultados de los años del 2007 al 1999 (años que no aparecen en los resultados de la consulta anterior).

Al consultar la selección en los dos casos, se obtendrán dos tablas con las mismas columnas que ofrecen la posibilidad de ser descargadas en distintos tipos.

		Descargar							
		Descargar la tabla en formato:							
		Pc-Axis Excel: extensión XLS Excel: extensión XLSx CSV separado por tabuladores CSV separado por , CSV separado por ;							
	Nacional								
	Total CNAE								
	Total								
2016	3.236.582								161.364
2015	3.186.678								161.119
2014	3.119.310								159.473
2013	3.146.570	471.521	88.067	66.869	85.044	129.688	37.190		162.153
2012	3.199.617	482.334	89.116	66.967	85.372	131.315	38.137		164.994
2011	3.250.576	492.341	90.858	69.877	87.461	132.488	38.867		166.509
2010	3.291.263	498.579	92.205	70.362	89.562	135.964	39.024		168.972
2009	3.355.830	510.072	93.283	71.853	91.826	139.381	39.611		170.626
2008	3.422.239	522.815	94.931	73.124	93.335	143.471	40.393		173.209

**Figura 41:** Descargar resultados obtenidos de la consulta (Fuente: INEBase-DIRCE)

Seleccionaremos la opción de 'CSV separado por;' y se descargará un archivo .csv con toda la información estructurada de la siguiente manera:

```

ExplotaciUn Estadística del Directorio Central de Empresas
Resultados por comunidades autónomas
Empresas por CCAA, actividad principal (grupos CNAE 2009) y estrato de asalariados.
Unidades: Empresas

;Nacional;Andalucía;Aragón;Asturias, Principado de;Balears, Illes;Canarias;Cantabria;Castilla y León;Castilla - La
Mancha;Cataluña;Comunitat Valenciana;Extremadura;Galicia;Madrid, Comunidad de;Murcia, Región de;Navarra, Comunidad Foral
de;País Vasco;Rioja, La;Ceuta;Melilla;
;Total CNAE;Total CNAE;Total CNAE;Total CNAE;Total CNAE;Total CNAE;Total CNAE;Total CNAE;Total CNAE;Total CNAE;Total
CNAE;Total CNAE;Total CNAE;Total CNAE;Total CNAE;Total CNAE;Total CNAE;Total CNAE;Total CNAE;Total CNAE;
;Total;Total;Total;Total;Total;Total;Total;Total;Total;Total;Total;Total;Total;Total;Total;Total;Total;Total;Total;Total;
2016;3236582.0;489347.0;90325.0;67675.0;89341.0;135909.0;37696.0;161364.0;126421.0;596196.0;344556.0;65484.0;197538.0;51641
2.0;92008.0;43907.0;151216.0;23076.0;3762.0;4349.0;
2015;3186678.0;480094.0;89873.0;67451.0;87111.0;132857.0;37332.0;161119.0;125786.0;584369.0;339533.0;64475.0;195595.0;50861
2.0;90031.0;43131.0;148575.0;23083.0;3749.0;4102.0;
2014;3119310.0;468930.0;88114.0;66342.0;84270.0;128518.0;36698.0;159473.0;123095.0;576565.0;330855.0;62929.0;191745.0;49450
9.0;86782.0;41582.0;149245.0;22314.0;3590.0;3754.0;
2013;3146570.0;471521.0;88067.0;66869.0;85044.0;129566.0;37190.0;162153.0;124405.0;580804.0;337161.0;63353.0;192998.0;49600
3.0;87146.0;40860.0;153709.0;22316.0;3610.0;3795.0;
2012;3199617.0;482334.0;89116.0;68967.0;85372.0;131315.0;38137.0;164994.0;127632.0;592192.0;342484.0;64671.0;194511.0;49909
8.0;88606.0;41305.0;159005.0;22486.0;3622.0;3770.0;
2011;3250576.0;492341.0;90858.0;69877.0;87461.0;132488.0;38867.0;166509.0;130079.0;601801.0;348955.0;65103.0;196535.0;50166
9.0;90856.0;41541.0;165496.0;22801.0;3642.0;3697.0;
2010;3291263.0;498579.0;92205.0;70362.0;89562.0;135954.0;39024.0;168972.0;131836.0;609670.0;352366.0;65573.0;198874.0;50350
1.0;92196.0;42347.0;169782.0;23190.0;3615.0;3655.0;
2009;3355830.0;510072.0;93283.0;71853.0;91826.0;139381.0;39611.0;170626.0;134479.0;619624.0;362844.0;67181.0;201263.0;51180
4.0;95636.0;43282.0;172152.0;23525.0;3689.0;3699.0;
2008;3422239.0;522815.0;94931.0;73124.0;93335.0;143471.0;40393.0;173209.0;137823.0;626020.0;376093.0;67852.0;203374.0;51930
7.0;100075.0;43847.0;175303.0;23834.0;3724.0;3709.0;
Notas:
Fuente: Instituto Nacional de Estadística

```

**Figura 42:** Archivo .csv descargado (Fuente: INEBase-DIRCE)

Este fichero tal cual ha sido descargado no puede ser interpretado por la herramienta que phpmAdmin proporciona para importar tablas. Por tanto, el archivo será modificado eliminando la cabecera, las filas con Total CNAE y Total y el final del fichero. Además, como los resultados que se mostraran en nuestra herramienta están en función de los años, se ha añadido una nueva columna al inicio de todas las demás denominada año en el que se han introducido los años que le corresponden los datos de cada fila. El resultado final del archivo csv es el siguiente:



## Creación de una herramienta para la catalogación de empresas, orientada a estudios de gestión

```
Año; Nacional; Andalucía; Aragón; Asturias, Principado de; Balears, Illes; Canarias; Cantabria; Castilla y León; Castilla - La Mancha; Cataluña; Comunitat Valenciana; Extremadura; Galicia; Madrid, Comunidad de; Murcia, Región de; Navarra, Comunidad Foral de; País Vasco; Rioja, La; Ceuta; Melilla
2016; 3236582.0; 489347.0; 90325.0; 67675.0; 89341.0; 135909.0; 37696.0; 161364.0; 126421.0; 596196.0; 344556.0; 65484.0; 197538.0; 516412.0; 92008.0; 43907.0; 151216.0; 23076.0; 8111.0
2015; 3186878.0; 480094.0; 89873.0; 67451.0; 87111.0; 132857.0; 37332.0; 161119.0; 125786.0; 584369.0; 339533.0; 64475.0; 195595.0; 508612.0; 90031.0; 43131.0; 148575.0; 23083.0; 7851.0
2014; 3119310.0; 468930.0; 88114.0; 66342.0; 84270.0; 128518.0; 36698.0; 159473.0; 123095.0; 576565.0; 330855.0; 62929.0; 191745.0; 494509.0; 86782.0; 41582.0; 149245.0; 22314.0; 7344.0
2013; 3146570.0; 471521.0; 88067.0; 66869.0; 85044.0; 129566.0; 37190.0; 162153.0; 124405.0; 580804.0; 337161.0; 63353.0; 192998.0; 496003.0; 87146.0; 40860.0; 153709.0; 22316.0; 7405.0
2012; 3199617.0; 482334.0; 89116.0; 68967.0; 85372.0; 131315.0; 38137.0; 164994.0; 127632.0; 592192.0; 342484.0; 64671.0; 194511.0; 499098.0; 88606.0; 41305.0; 159005.0; 22486.0; 7392.0
2011; 3250576.0; 492341.0; 90858.0; 69877.0; 87461.0; 132488.0; 38867.0; 166509.0; 130079.0; 601801.0; 348955.0; 65103.0; 196535.0; 501669.0; 90856.0; 41541.0; 165496.0; 22801.0; 7339.0
2010; 3291263.0; 498579.0; 92205.0; 70362.0; 89562.0; 135954.0; 39024.0; 168972.0; 131836.0; 609670.0; 352366.0; 65573.0; 198874.0; 503501.0; 92196.0; 42347.0; 169782.0; 23190.0; 7270.0
2009; 3355830.0; 510072.0; 93283.0; 71853.0; 91826.0; 139381.0; 39611.0; 170626.0; 134479.0; 619624.0; 362844.0; 67181.0; 201263.0; 511804.0; 95636.0; 43282.0; 172152.0; 23525.0; 7388.0
2008; 3422239.0; 522815.0; 94931.0; 73124.0; 93335.0; 143471.0; 40393.0; 173209.0; 137823.0; 626020.0; 376093.0; 67852.0; 203374.0; 519307.0; 100075.0; 43847.0; 175303.0; 23834.0; 4433.0
2007; 3336657.0; 511728.0; 92162.0; 72276.0; 91254.0; 140414.0; 39560.0; 170319.0; 132906.0; 612404.0; 368586.0; 66232.0; 200020.0; 503000.0; 97374.0; 43142.0; 164431.0; 23404.0; 7445.0
2006; 3174393.0; 486674.0; 87941.0; 70115.0; 88027.0; 132810.0; 37690.0; 163856.0; 124413.0; 578340.0; 348692.0; 63084.0; 191642.0; 478202.0; 90698.0; 41083.0; 161376.0; 22393.0; 7357.0
2005; 3064129.0; 464179.0; 90005.0; 68175.0; 87024.0; 128020.0; 36561.0; 159196.0; 118396.0; 567019.0; 329334.0; 61898.0; 185722.0; 456175.0; 85110.0; 40730.0; 157539.0; 21598.0; 7448.0
2004; 2942583.0; 441623.0; 85814.0; 67039.0; 85425.0; 120294.0; 35649.0; 155004.0; 113967.0; 543719.0; 315214.0; 55568.0; 180977.0; 436074.0; 82484.0; 39679.0; 155666.0; 21049.0; 7338.0
2003; 2813159.0; 417006.0; 82750.0; 65062.0; 75951.0; 117871.0; 34017.0; 151448.0; 107294.0; 525557.0; 299452.0; 58306.0; 174077.0; 407655.0; 75973.0; 38936.0; 154703.0; 20094.0; 7007.0
2002; 2710400.0; 398302.0; 81533.0; 63209.0; 73358.0; 113165.0; 33017.0; 147496.0; 100492.0; 515173.0; 288408.0; 57419.0; 165726.0; 384638.0; 72547.0; 37333.0; 152211.0; 19499.0; 6874.0
2001; 2645317.0; 384086.0; 80586.0; 65301.0; 71424.0; 108619.0; 32288.0; 147775.0; 100804.0; 494153.0; 281932.0; 47765.0; 167732.0; 377785.0; 69865.0; 36606.0; 152722.0; 18985.0; 6889.0
```

**Figura 43:** Archivo .csv modificado para que sea interpretado por phpmyAdmin  
(Fuente: elaboración propia)

Una vez tenemos el archivo preparado, lo importamos desde el gestor de base de datos phpmyAdmin en nuestra base de datos creada:

Servidor: localhost » Base de datos: TFG

Estructura SQL Buscar Generar una consulta Exportar Importar Operaciones Privilegios

El archivo puede ser comprimido (gzip, bzip2, zip) o descomprimido.  
Un archivo comprimido tiene que terminar en **[formato].[compresión]**. Por ejemplo: **.sql.zip**

Buscar en su ordenador:  CCAA(2016...008).csv (Máximo: 128MB)  
También puede arrastrar un archivo en cualquier página.

Conjunto de caracteres del archivo:

---

**Importación parcial:**

Permitir la interrupción de una importación en caso que el script detecte que se ha acercado al límite de tiempo PHP. (Esto podría ser un buen método, pero puede dañar las transacciones.)

Omitir esta cantidad de consultas (en SQL) desde la primera:

---

**Otras opciones:**

Habilite la revisión de las claves foráneas

---

**Formato:**

*Nota: si la fila contiene múltiples tablas, van a ser combinadas en una.*

---

**Opciones específicas al formato:**

Actualizar datos cuando las llaves importadas están duplicadas (agregar ON DUPLICATE KEY UPDATE)

Columnas separadas por:

Columnas encerradas entre:

Caracter de escape de columnas:

Líneas terminadas en:

La primer línea del archivo contiene los nombres de columna de la tabla (si no está activado la primera línea será parte de los datos)

Consola

**Figura 44:** Formulario de importación del archivo .csv (Fuente: phpmyAdmin)

Con ello, ya tendremos la tabla creada con los datos del año 2016 al 2008. Para añadir la información que se corresponde a los años 2007 al 1999 se seguirá el mismo procedimiento, pero en vez de en una tabla nueva la importación se realizará sobre la tabla creada. También se seguirá el mismo proceso para crear el resto de tablas de nuestra base de datos.



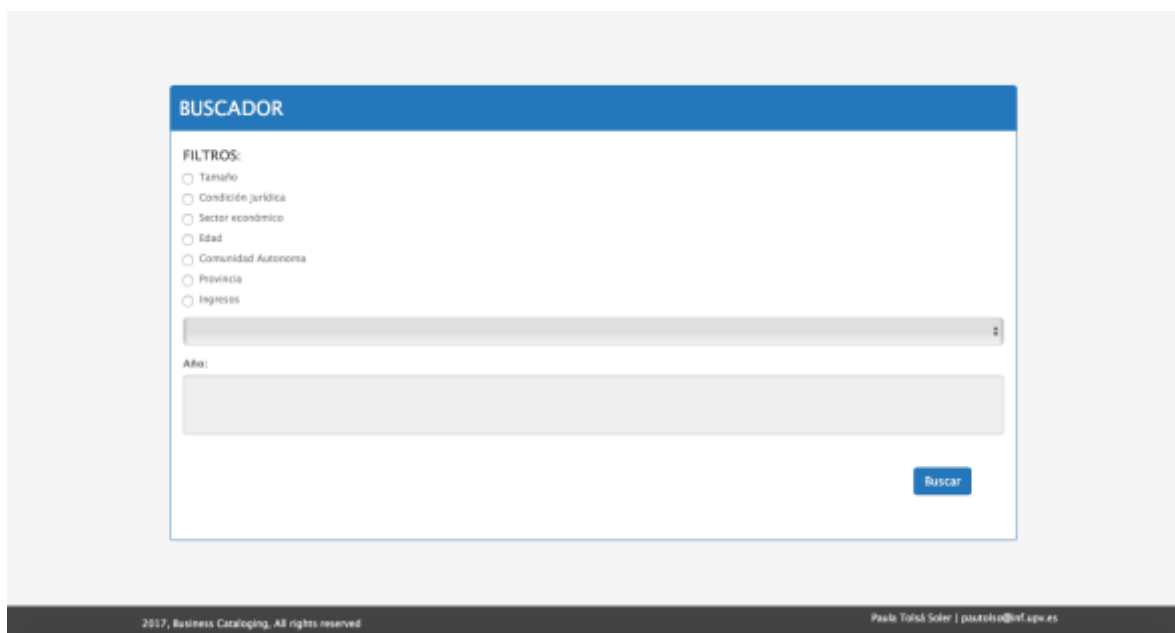
### 7.3. Manual de la herramienta

Como parte final, se hará un breve repaso de cómo funciona la herramienta, a través de capturas de pantalla:

Al acceder a la URL <http://localhost/TFGProject/> nos encontramos con la siguiente pantalla:



*Figura 45: Página inicial*



*Figura 46: Página inicial*



Se trata de un formulario que nos permite buscar datos de la población empresarial según su tamaño, condición jurídica, sector económico, Edad, Comunidad Autónoma, Provincia o Ingresos.

Los datos que se obtienen de la búsqueda pueden ser generales, en el caso de que accionemos sobre el botón buscar habiendo determinado únicamente un filtro. Pero pueden ser más concretos gracias al desplegable que se carga en el formulario inicial cada vez que se selecciona uno de los filtros con las distintas modalidades del filtro. Y gracias a un contenedor con los años disponibles que pueden también ser seleccionados.

Entendemos por modalidad las distintas variantes del filtro, por ejemplo, para el filtro sector económico, sus modalidades serán: comercio, industria, construcción y servicios. O para el filtro Provincia, sus modalidades serán cada una de las provincias.

Por tanto, dependiendo de seleccionado en este formulario, se obtendrán unos datos u otros. Estos resultados podrán ser visualizados o bien mediante una tabla o mediante gráficas. Al igual de que se mostrarán una tabla de comparaciones respecto a los datos de un año base. Estas tres opciones pueden ser elegidas en la misma página mediante un elemento de navegación por pestañas el cual hace que el contenido a mostrar cambie dinámicamente al pulsar sobre cada una de ellas.

**BUSCADOR**

RESULTADOS

CATALOGACIÓN POR COMUNIDAD AUTÓNOMA [◀ Volver al buscador](#)

Tabla Gráficas Variaciones

Año	Nacional	Andalucía	Aragón	Asturias, Principado de	Balears, Illes	Canarias	Cantabria	Castilla y León	Castilla - La Mancha	Cataluña	Comunitat Valenciana	E
2016	3236582.0	489347.0	90125.0	67675.0	89341.0	135909.0	37696.0	161364.0	126421.0	596196.0	344356.0	
2015	3186678.0	480094.0	89873.0	67451.0	87111.0	132857.0	37332.0	161119.0	125786.0	584360.0	339533.0	
2014	3119310.0	468930.0	88114.0	66342.0	84270.0	128518.0	36698.0	159473.0	123095.0	576565.0	330855.0	
2013	3146570.0	471521.0	88067.0	66869.0	85044.0	129566.0	37190.0	162153.0	124405.0	580804.0	337361.0	
2012	3199617.0	482334.0	89116.0	68967.0	85372.0	131315.0	38137.0	164994.0	127632.0	592192.0	342484.0	
2011	3250570.0	492341.0	90858.0	69877.0	87461.0	132488.0	38867.0	166509.0	130079.0	601801.0	348955.0	
2010	3291263.0	498579.0	92205.0	70362.0	89562.0	135954.0	39024.0	168972.0	131836.0	609670.0	352166.0	
2009	3355830.0	510072.0	93283.0	71853.0	91826.0	139381.0	39611.0	170626.0	134479.0	619624.0	362844.0	
2008	3422239.0	522815.0	94931.0	73124.0	93335.0	143471.0	40393.0	173209.0	137823.0	626020.0	370093.0	
2007	3336657.0	511728.0	92182.0	72276.0	91254.0	140414.0	39560.0	170319.0	132906.0	612404.0	368586.0	
2006	3174393.0	486674.0	87941.0	70115.0	88027.0	132810.0	37690.0	163856.0	124413.0	578340.0	348692.0	
2005	3084129.0	464179.0	90005.0	88175.0	87024.0	128020.0	36361.0	159196.0	118396.0	567019.0	329334.0	
2004	2942583.0	441623.0	85814.0	67039.0	85425.0	120294.0	35649.0	155004.0	113967.0	543719.0	315214.0	
2003	2813159.0	417006.0	82750.0	65062.0	75951.0	117871.0	34017.0	151448.0	107294.0	525557.0	299452.0	

Figura 47: Resultados en forma de tabla

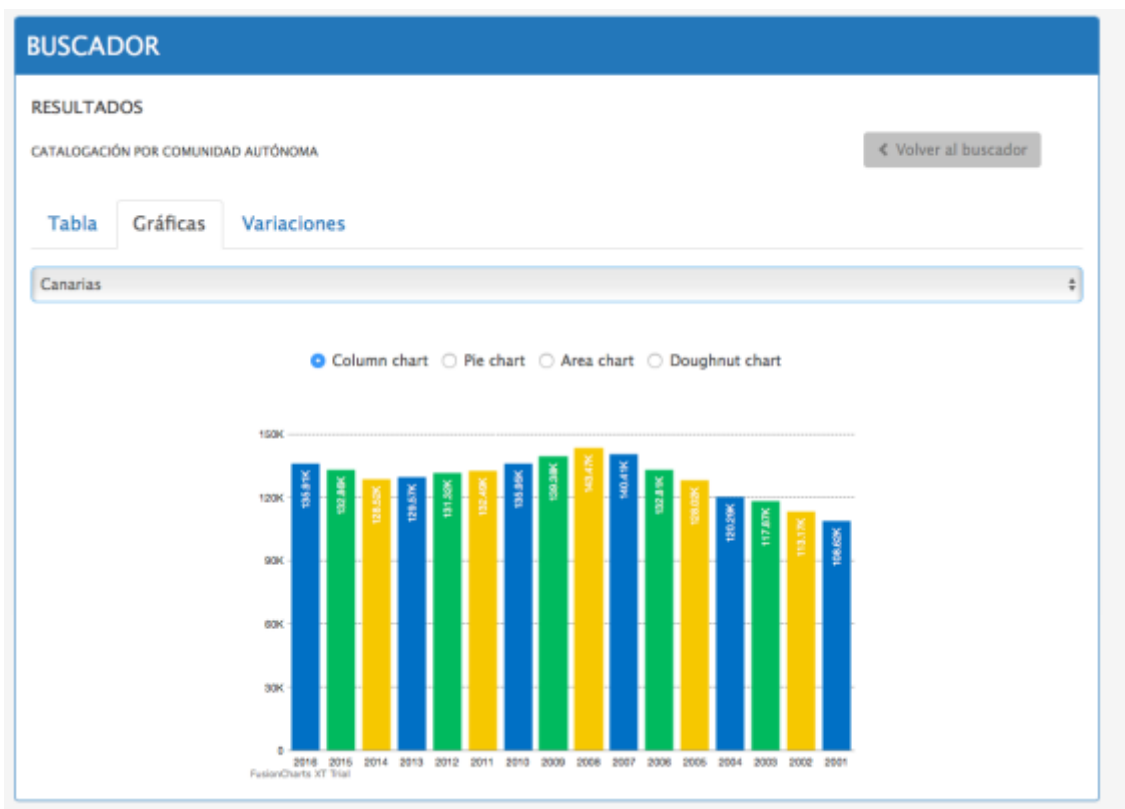


Figura 48: Resultados en forma de gráfica en columnas

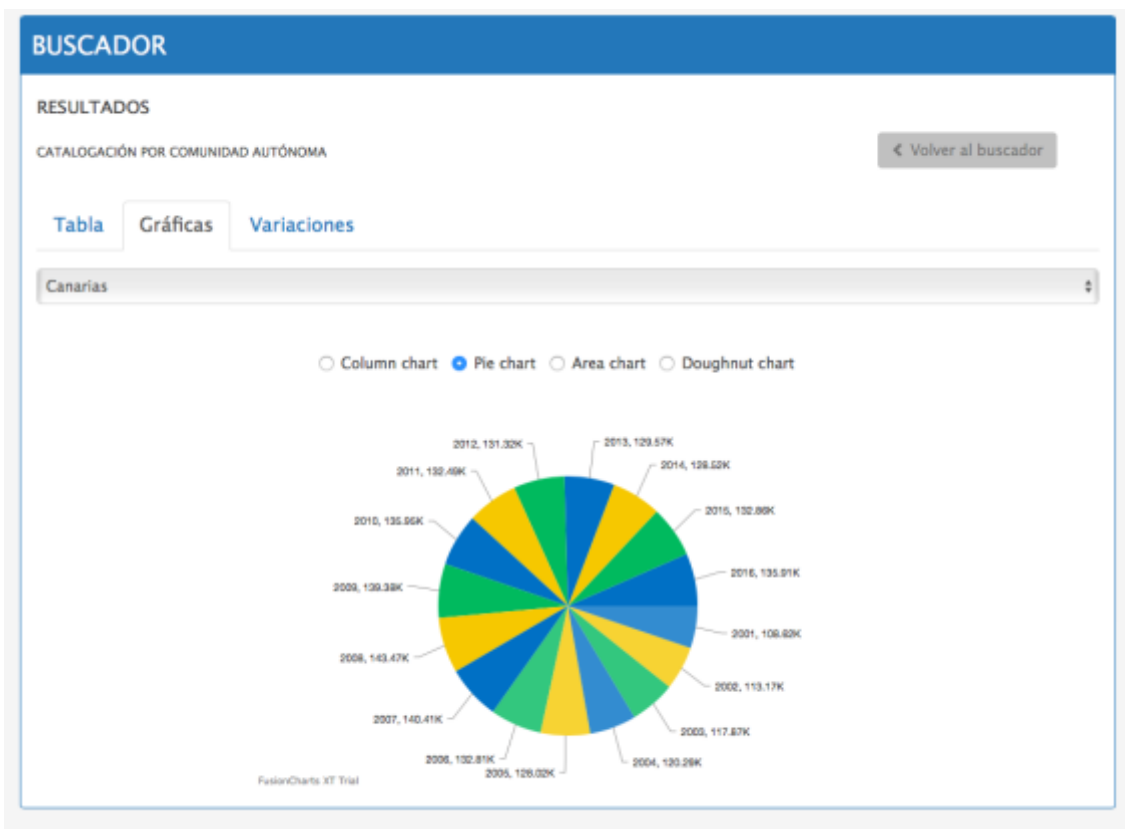


Figura 49: Resultados en forma de 'gráfica de pastel'

Tabla Gráficas Variaciones

Canarias  
Canarias

Año base:  
2016

NÚMERO DE EMPRESAS EN EL AÑO 2016: 135909.0

Año	Empresas	Diferencia con respecto al 2016	Diferencia %
2015	132857.0	3052	2.3%
2014	128518.0	7391	5.75%
2013	129566.0	6343	4.9%
2012	131315.0	4594	3.5%
2011	132488.0	3421	2.58%
2010	135954.0	-45	-0.03%
2009	139381.0	-3472	-2.49%
2008	143471.0	-7562	-5.27%
2007	140414.0	-4505	-3.21%
2006	132810.0	3099	2.33%
2005	128020.0	7889	6.16%
2004	120294.0	15615	12.98%
2003	117871.0	18038	15.3%
2002	113165.0	22744	20.1%
2001	108619.0	27290	25.12%

**Figura 50:** Tabla de variaciones con respecto al año base 2016



## 8. Bibliografía

---

- [1] Metodología DIRCE - *ine.es*  
<http://www.ine.es/metodologia/t37/t3730200.pdf>
  
- [2] Documento DIRCE 2016 - *ine.es*  
<http://www.ine.es/daco/daco42/dirce/dirce16.pdf>
  
- [3] INE ,2017- *ine.es*  
[http://www.ine.es/ss/Satellite?L=es\\_ES&c=Page&cid=1254735905566&p=1254735905566&pagename=INE%2FINELayout](http://www.ine.es/ss/Satellite?L=es_ES&c=Page&cid=1254735905566&p=1254735905566&pagename=INE%2FINELayout)
  
- [4] INEbase / ¿Qué es INEbase? - *ine.es*  
<http://www.ine.es/inebmenu/queesinebase.htm>
  
- [5] Instituto Nacional de Estadística (España), 2017. *Es.wikipedia.org*  
[https://es.wikipedia.org/wiki/Instituto\\_Nacional\\_de\\_Estadística\\_\(España\)](https://es.wikipedia.org/wiki/Instituto_Nacional_de_Estadística_(España))
  
- [6] Estructura y dinamismo del tejido empresarial en España. Directorio Central de Empresas (DIRCE) a 1 de enero de 2016 - *ine.es*  
<http://www.ine.es/prensa/np984.pdf>
  
- [7] Ciclo de vida del software - *EcuRed*  
[https://www.ecured.cu/Ciclo\\_de\\_vida\\_del\\_software](https://www.ecured.cu/Ciclo_de_vida_del_software)
  
- [8] Ciclo de vida del 'software'- *CCM*  
<http://es.ccm.net/contents/223-ciclo-de-vida-del-software>
  
- [9] Desarrollo en cascada- *Es.wikipedia.org*  
[https://es.wikipedia.org/wiki/Desarrollo\\_en\\_cascada](https://es.wikipedia.org/wiki/Desarrollo_en_cascada)



- [10] Instituto Nacional de Estadística. (National Statistics Institute), DIRCE - *Ine.es*

<http://www.ine.es/dynt3/inebase/index.htm?padre=51>

- [11] Área de estadísticas de la Dirección General de Política de la Pequeña y Mediana Empresa - *Estadisticas.ipyme.org*

<http://estadisticas.ipyme.org/Empresas/Empresas.aspx>

- [12] Conjuntos de datos - *datos.gob.es*

[http://datos.gob.es/es/catalogo?publisher\\_display\\_name=Instituto+Nacional+de+Estadística&administration\\_level=E&theme\\_id=economia](http://datos.gob.es/es/catalogo?publisher_display_name=Instituto+Nacional+de+Estadística&administration_level=E&theme_id=economia)

- [13] Web estática vs web dinámica - *Artfactory.es*

<http://www.artfactory.es/es/news/web-estatica-vs-web-dinamica>

- [14] Páginas web ventajas y desventajas- *Es.slideshare.net*

[https://es.slideshare.net/angelaelizabethsamaniegoorozco/paginas-web-ventajas-y-desventajas?next\\_slideshow=1](https://es.slideshare.net/angelaelizabethsamaniegoorozco/paginas-web-ventajas-y-desventajas?next_slideshow=1)

- [15] Cómo crear un sitio web con secciones dinámicas con HTML, PHP, Delphi y MySQL Proyecto AjpdSoft- *Ajpdsoft.com*

<http://www.ajpdsoft.com/modules.php?name=News&file=article&sid=488>

- [16] Base de datos-*Es.wikipedia.org*

[https://es.wikipedia.org/wiki/Base\\_de\\_datos](https://es.wikipedia.org/wiki/Base_de_datos)

- [17] ¿Qué son las bases de datos? - *Maestros del Web*

<http://www.maestrosdelweb.com/que-son-las-bases-de-datos/>

- [18] MySQL- *Es.wikipedia.org*

<https://es.wikipedia.org/wiki/MySQL#Caracter.C3.ADsticas>

- [19] Diseño Web: 15 Principios Elementales de Interfaz de Usuario - *VegaCorp*  
Autor: Jose Vega

<http://es.vegacorp.me/disenio-web-principios-elementales-interfaz-usuario/>

[20] Interfaz gráfica de usuario - *Es.wikipedia.org*

[https://es.wikipedia.org/wiki/Interfaz gráfica de usuario](https://es.wikipedia.org/wiki/Interfaz_gráfica_de_usuario)

[21] HTML Tutorial - *W3schools.com*

<http://www.w3schools.com/html/>

[22] HTML-*Es.wikipedia.org*

<https://es.wikipedia.org/wiki/HTML>

[23] ¿Que es HTML y para qué sirve? - *Acerca de HTML*

<http://www.acercadehtml.com/manual-html/que-es-html.html>

[24] Guía Breve de CSS - *W3c.es*

<http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>

[25] Hoja de estilos en cascada- *Es.wikipedia.org*

[https://es.wikipedia.org/wiki/Hoja\\_de\\_estilos\\_en\\_cascada](https://es.wikipedia.org/wiki/Hoja_de_estilos_en_cascada)

[26] CSSTutorial - *W3schools.com*

<http://www.w3schools.com/css/>

[27] Bootstrap - The world's most popular mobile-first and responsive front-end framework - *Getbootstrap.com*

<http://getbootstrap.com>

[28] Bootstrap (framework)- *Es.wikipedia.org*

[https://es.wikipedia.org/wiki/Bootstrap \(framework\)](https://es.wikipedia.org/wiki/Bootstrap_(framework))

[29] HTML Snippets for Twitter Bootstrap framework - *Bootsnipp.com*

<https://bootsnipp.com/tags>



- [30] JavaScript - *Es.wikipedia.org*  
<https://es.wikipedia.org/wiki/JavaScript>
- [31] JavaScript Tutorial - *W3schools.com*  
<http://www.w3schools.com/js/>
- [32] JavaScript-*Mozilla Developer Network*  
<https://developer.mozilla.org/es/docs/Web/JavaScript>
- [33] The jQuery Foundation. (2016). jQuery API.  
<http://api.jquery.com/>
- [34] Javascript a fondo - *Desarrolloweb.com*  
<https://desarrolloweb.com/javascript/#quees>
- [35] *Blog.udemy.com*  
<https://blog.udemy.com/jquery-vs-javascript-2-cual-es-la-diferencia-en-definitiva/>
- [36] ¿Como Conectar Php y Mysql a través de PDO y MySQLi?- Hermosa Programación: +50 Tutoriales Desarrollo Android  
Autor: James Revelo  
<http://www.hermosaprogramacion.com/2014/07/php-mysql-conectar-como/>
- [37] PHP: mysqli::set\_charset - Manual - *Php.net*  
<http://php.net/manual/es/mysqli.set-charset.php>
- [38] Página oficial PHP  
<http://www.php.net/>
- [39] MySQL | La base de datos de código abierto más popular - *Oracle.com*



<https://www.oracle.com/lad/mysql/index.html>

[40] Temario: Introducción. MySql - *Tutorialesprogramacionya.com*

<http://www.tutorialesprogramacionya.com/mysqlya/temarios/descripcion.php?inicio=0&cod=2&punto=1>

[41] Qué es MySQL? :: esepestudio, especialistas web - *Esepestudio.com*

<http://www.esepestudio.com/noticias/que-es-mysql>

[42] NetBeans- *Es.wikipedia.org*

<https://es.wikipedia.org/wiki/NetBeans>

[43] NetBeans- *Genbetadev.com*

<https://www.genbetadev.com/herramientas/netbeans-1>

[44] XAMPP Installers and Downloads for Apache Friends - *Apachefriends.org*

<https://www.apachefriends.org/es/index.html>

[45] XAMPP- *Es.wikipedia.org*

<https://es.wikipedia.org/wiki/XAMPP>

[46] Formularios básicos en HTML - *Htmlquick.com*

Autor: Diego León

<http://www.htmlquick.com/es/tutorials/forms.html#concept>

[47] Elementos de navegación (Bootstrap 3, el manual oficial) - *Librosweb.es*

[http://librosweb.es/libro/bootstrap\\_3/capitulo\\_6/elementos\\_de\\_navegacion.html](http://librosweb.es/libro/bootstrap_3/capitulo_6/elementos_de_navegacion.html)

[48] JavaScript Charts for Web, Mobile & Apps - FusionCharts - *FusionCharts*

<http://www.fusioncharts.com>

[49] FusionCharts Suite XT- *Danysoft : Soluciones Software Profesionales*



<http://www.danysoft.com/fusioncharts-suite-xt/>

[50] Guía Breve sobre Estándares Web. - *W3schools.com*

<http://www.w3c.es/Divulgacion/GuiasBreves/Estandares>

[51] Apuntes de la asignatura ISW del grado en Ingeniería Informática de la UPV.

[52] The W3C CSS Validation Service- *Jigsaw.w3.org*

<https://jigsaw.w3.org/css-validator/>