



Universidad
de La Laguna

Escuela Superior de
Ingeniería y Tecnología
Sección de Ingeniería Informática

Trabajo de Fin de Grado

VIStac-IMAS-Can

Eliasib Jesús García Martín

La Laguna, 7 de julio de 2015

D^a. **Luz Marina Moreno De Antonio**, con N.I.F. 45.457.492-Q profesora adscrita al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutora

D^a. **Isabel Sánchez Berriel**, con N.I.F. 42.885.838-S profesora adscrita al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como cotutora

C E R T I F I C A N

Que la presente memoria titulada:

“VISTac-IMAS-Can”

ha sido realizada bajo su dirección por D. **Eliasib Jesús García Martín**, con N.I.F. 78.649.884-G.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 7 de julio de 2015.

Agradecimientos

A mi familia y amigos,

A Luz Marina Moreno De Antonio e Isabel Sánchez Berriel,

Por su ayuda y labor docente en el desarrollo de este proyecto

*Y a Rafael Betancor Villalba por su colaboración e implicación en
la disponibilidad de las fuentes de trabajo.*

Licencia



© Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial 4.0 Internacional.

Resumen

El objetivo de este trabajo ha sido la creación de una aplicación móvil multiplataforma que permita visualizar datos estadísticos recogidos por el Instituto Canario de Estadística (ISTAC) relacionados con el Área de Medioambiente y Sostenibilidad del Gobierno de Canarias. En concreto se ha obtenido versiones para los sistemas operativos móviles Android e iOS. Los indicadores estadísticos se reciben mediante peticiones a una API o mediante la combinación y transformación de varios indicadores básicos. La representación de los datos es tanto numérica como gráfica.

Palabras clave: Aplicación móvil multiplataforma, Visualización de datos, indicadores estadísticos, SIMAC, ISTAC.

Abstract

The main goal of this project is the development of a multiplatform mobile application in which visualize statistics data collected by the Canary Islands Statistics Institute (ISTAC) related with the Environment and Sustainability Area of the Canary Islands Government. Specific versions for both Android and iOS mobile operating systems have been developed. Statistics indicators are received through API requests or combining and transforming some basic indicators. Data representation is both numerical and graphical.

Keywords: *Multiplatform mobile application, data visualization, statistics indicators, SIMAC, ISTAC.*

Índice General

Parte I. Introducción y fundamentos básicos	8
Capítulo 1. Introducción	9
1.1 Objetivos.....	9
1.2 Alcance.....	9
Capítulo 2. Estado del arte	11
2.1 Estado de la tecnología.....	11
2.1.1 Introducción a <i>frameworks</i> de desarrollo de aplicaciones	11
2.1.2 Tecnologías a utilizar en el proyecto	12
2.1.3 API ISTAC.....	14
2.2 Antecedentes software	15
Parte II. Cuerpo principal. Descripción del trabajo	18
Capítulo 3. Indicadores económicos	19
3.1 Introducción.....	19
3.2 SIMAC.....	19
3.3 Selección de indicadores.....	20
Capítulo 4. Visualización de datos	24
4.1 Introducción a la visualización de datos	24
4.2 Formato de los datos.....	24
4.3 Técnicas de visualización de datos.....	25
4.3.1 Gráficos de barras	26
4.3.2 Gráficos de líneas	26
4.3.3 Gráficos circulares o de sectores.....	27
4.4 Características deseables.....	28

4.5	Servicios de visualización.....	32
4.5.1	Aplicaciones	32
4.5.2	Librerías.....	32
Capítulo 5.	Desarrollo	34
5.1	Plan de trabajo.....	34
5.2	Tecnologías empleadas.....	34
5.2.1	Petición 1: Inicialización de la aplicación	35
5.2.2	Petición 2: Categorías	36
5.2.3	Petición 3: Indicadores de la categoría seleccionada	36
5.2.4	Petición 4: Datos del indicador para el filtro de consulta	37
5.2.5	Petición 5: Datos estadísticos del indicador.....	38
5.3	Herramientas.....	39
5.4	Tratamiento de la información	40
5.5	Funcionamiento.....	44
5.6	Problemas encontrados.....	48
5.6.1	Cross-Domain.....	48
Capítulo 6.	Pruebas de usabilidad	50
6.1	Motivación	50
6.2	Pruebas realizadas.....	51
6.3	Resultados obtenidos.....	52
6.4	Propuesta de cambios.....	53
6.4.1	Cambios en la disposición de elementos	53
6.4.2	Cambios para la reducción de elementos intrusivos	54
6.4.3	Mejoras en el formato numérico	54
6.4.4	Mejoras en la selección de opciones del filtro de consulta	54
Parte III.	Conclusiones	56

Capítulo 7. Conclusiones y líneas futuras	57
Capítulo 8. Summary and Conclusions	59
Capítulo 9. Presupuesto	60
Apéndice A. Creación de gráficas con D3	62
A.1. Ejemplo: Gráfico de líneas.....	62
Bibliografía	65

Índice de figuras

Figura 2.1 Ejemplo de petición y respuesta en la herramienta Swagger	14
Figura 2.2 Pantalla de inicio de PULSEC.....	16
Figura 2.3 Resultados de una consulta en PULSEC.....	16
Figura 2.4 Página de consulta de indicadores económicos del EUROSTAT ..	17
Figura 2.5 Página de obtención de <i>widjets</i> del ISTAC.....	17
Figura 4.1 Ejemplo de gráfico de barras [Fuente: Snapsurveys].....	26
Figura 4.2 Ejemplo de gráfico de barras [Fuente: IndiaBIX]	27
Figura 4.3 Ejemplo de gráfico circular o de sectores [Fuente: ADR Formación]	28
Figura 4.4 Ejemplo de salida de los datos numéricos de una consulta.....	29
Figura 4.5 Ejemplo de salida de los datos gráficos para las islas de una consulta	31
Figura 4.6 Ejemplo de salida de los datos gráficos para Canarias de una consulta	31
Figura 5.1 Diagrama de Petición 1: Inicialización	35
Figura 5.2 Diagrama de Petición 2: Categorías.....	36
Figura 5.3 Diagrama de Petición 3: Indicadores.....	36
Figura 5.4 Diagrama de Petición 4: Parámetros geográficos y temporales de un indicador básico.....	37
Figura 5.5 Diagrama de Petición 4: Parámetros geográficos y temporales de un indicador derivado.....	37
Figura 5.6 Diagrama de Petición 5: Datos estadísticos de un indicador básico	38
Figura 5.7 Diagrama de Petición 5: Datos estadísticos de un indicador derivado	38
Figura 5.8 Pantalla de categorías.....	44
Figura 5.9 Indicadores de la categoría “Población”	45

Figura 5.10 Islas y años disponibles para el indicador “Población. De 15 a 64 años”	45
Figura 5.11 Selección de criterios de búsqueda.....	46
Figura 5.12 Botón “Realizar consulta”	46
Figura 5.13 Resultados numéricos de la consulta	47
Figura 5.14 Resultados gráficos de la consulta.....	47

Índice de tablas

Tabla 2.1 Comparación de librerías de manipulación DOM.....	13
Tabla 3.1 Selección de indicadores.....	23
Tabla 5.1 Distribución temporal del plan de trabajo.....	34
Tabla 5.2 Estado final de las listas en Trello	40
Tabla 9.1 Estimación temporal del desarrollo del proyecto	60
Tabla 9.2 Presupuesto total del proyecto.....	61

Parte I.

Introducción y fundamentos básicos

Capítulo 1.

Introducción

1.1 Objetivos

El objetivo principal de este proyecto es la creación de una aplicación móvil para los sistemas operativos Android e iOS que permita la consulta de indicadores estadísticos recogidos por el Instituto Canario de Estadística (ISTAC) para ser visualizados en forma numérica y gráfica. También se han incluido indicadores estadísticos obtenidos a partir de la aplicación de cálculos sobre ciertos indicadores básicos, para así ampliar la gama informativa de la aplicación.

Estos datos están relacionados con temas de interés para la Consejería de Medioambiente y Sostenibilidad del Gobierno de Canarias, y se han seleccionado entre los definidos en el Sistema de Indicadores Medioambientales de Canarias (SIMAC) [1].

Otro objetivo secundario ha sido la investigación del ámbito de uso de la aplicación, para así adaptar su contenido al que fuera el deseado para el público objetivo al que va dirigido esta aplicación.

1.2 Alcance

Se desarrollará una aplicación para Android e iOS que realice peticiones a la API del Instituto Canario de Estadística a través de la cual se visualicen datos referentes al área de Medioambiente y Sostenibilidad del Gobierno de Canarias.

Las peticiones se realizan tras la elección por parte del usuario de factores temporales y geográficos, como los años o las islas de los datos a consultar, recuperándose exclusivamente los datos correspondientes de interés, sin necesidad de descargar una base de datos completa para posteriormente ser filtrada.

Una vez recibidos los datos de la consulta, la visualización se realiza mostrando tanto los datos numéricos, como generando diversos tipos de gráficas, sobre las que resultará más cómodo analizar su variación a lo largo del tiempo.

Capítulo 2.

Estado del arte

2.1 Estado de la tecnología

2.1.1 Introducción a *frameworks* de desarrollo de aplicaciones

En la actualidad, existen diversos *frameworks* que permiten el desarrollo de aplicaciones móviles, escribiendo su código en lenguajes web tales como HTML5, CSS3 o JavaScript. De esta forma se diseña la aplicación como si se tratara de una página web, y posteriormente se puede exportar a diversos sistemas operativos móviles, entre los que se incluyen los dos más utilizados hoy en día: Android e iOS. También debe tenerse en cuenta la ventaja que supone el uso de lenguajes web para la creación de aplicaciones, dado el gran número de librerías con diversas funcionalidades disponibles para la creación de páginas web, e incluso, de las denominadas aplicaciones web.

La herramienta que se encarga del empaquetado del proyecto web para su exportación a diversas plataformas móviles se llama **Apache Cordova** [2], aunque al ser un producto de código abierto, se pueden crear productos basados en ella que incluyan mejoras desarrolladas por la comunidad. Un ejemplo es **Phonegap** [3] (Adobe), que básicamente es lo mismo que Apache Cordova, pero al pertenecer a una empresa pionera en el mundo del software ofrece diversos servicios en la web para dar soporte a las aplicaciones desarrolladas en Phonegap, así como un servicio online de exportación de aplicaciones a diversos sistemas operativos móviles. Una de las ventajas de este servicio es que podríamos exportar nuestra aplicación Phonegap a iOS sin necesidad de estar trabajando en un equipo Macintosh, requisito que sí debe cumplirse en caso de querer desarrollar una aplicación para iOS de forma nativa.

También existen otros tipos de *frameworks* para desarrollar aplicaciones, como por ejemplo **Appcelerator** [4], que permite escribir aplicaciones en JavaScript y un lenguaje de marcado propio similar a XML, con la desventaja de que sólo es útil para crear aplicaciones para Android, dejando de lado la

posibilidad de tener la aplicación también para iOS. Otro ejemplo es **React Native** [5], desarrollado por Facebook, del cual podría decirse que es el análogo de Appcelerator, pero para el sistema operativo iOS: permite crear aplicaciones nativas escritas en JavaScript y usa también ciertos aspectos de un lenguaje de marcado para crear los elementos visuales. Al igual que en el caso anterior, encontramos la desventaja de que sólo permite crear aplicaciones para iOS y no para Android, aunque sus creadores han anunciado que añadirán soporte para Android en un futuro.

2.1.2 Tecnologías a utilizar en el proyecto

El *framework* que se ha elegido para el desarrollo de este proyecto ha sido **Ionic Framework** [6], que permite crear aplicaciones para Android e iOS escritas en lenguajes web, como son HTML5, CSS3 y JavaScript. Dicho framework trabaja sobre la herramienta Apache Cordova, incorpora hojas de estilo propias que siguen los patrones de diseño más comunes de Android e iOS, y además incluye la librería **AngularJS** [7].

AngularJS es un framework de JavaScript de código abierto, mantenido por **Google**, que ayuda con la gestión de lo que se conoce como aplicaciones de una sola página; esto es, aquellas aplicaciones web que para navegar por ellas no precisan de una recarga completa de la página cada vez que se cambia el contenido de la misma, sino que los cambios se reflejan sólo en los elementos que sea necesario cambiar, evitando así la recarga innecesaria de elementos ya preparados en la aplicación. Esto se consigue gracias a la manipulación del árbol DOM de la aplicación. Gracias a AngularJS, la aplicación cumple el **modelo vista-controlador**, el cual es un patrón de diseño que permite sincronizar tanto los elementos de la interfaz gráfica de un programa (o vista) con el estado de los datos internos de la aplicación (o modelo), gracias a un intermediario que gestiona estos cambios, llamado controlador. AngularJS hace uso de etiquetas HTML personalizadas, en las cuales se pueden indicar mediante parámetros ciertas acciones a tener en cuenta por el controlador para que el elemento en cuestión reaccione ante un cambio en los datos o viceversa. En el caso de esta aplicación, resulta muy útil esta librería sobre todo en el aspecto de la implementación del MVC, dado que la disposición tanto de las categorías e indicadores, como de los formularios de búsqueda, y la disposición de los resultados siguen una plantilla que varía únicamente en función de los datos

recibidos, por lo que AngularJS gestiona perfectamente estas pequeñas variaciones entre los distintos datos.

Otra librería bastante común en un proyecto web son las que facilitan la manipulación del árbol DOM, para poder cambiar estos elementos de manera rápida, directa y sencilla. Entre este tipo de librerías, destaca **jQuery** [8], utilizada en infinidad de sitios web, y que es ya casi un estándar que toman como referencia otras librerías de propósito similar. En este proyecto, se utiliza la librería **Zepto.js** [9], que es una versión reducida de jQuery que incluye las funcionalidades más importantes de la librería original. A continuación, se muestra una tabla comparativa acerca de algunos tipos de funciones que soportan estas librerías:

Comparación de librerías de manipulación DOM		
Funciones	jQuery	Zepto.js
Selectores	Sí	Sí
Ajax	Sí	Sí
Eventos	Sí	Sí
Serialización de <i>arrays</i>	Sí	Sí
Utilidades (colas, copias, comprobación de tipos, etc.)	Sí	Sólo algunas
Estilos CSS	Sí	Sólo algunas
Objetos <i>diferidos</i>	Sí	No

Tabla 2.1 Comparación de librerías de manipulación DOM

Se ha escogido Zepto.js dado que jQuery puede llegar a resultar pesada en el desarrollo de una aplicación móvil, y tras analizar las funcionalidades de Zepto.js, se considera que son suficientes para el desarrollo de esta aplicación y que también es conveniente usarla para no sobrecargar la aplicación.

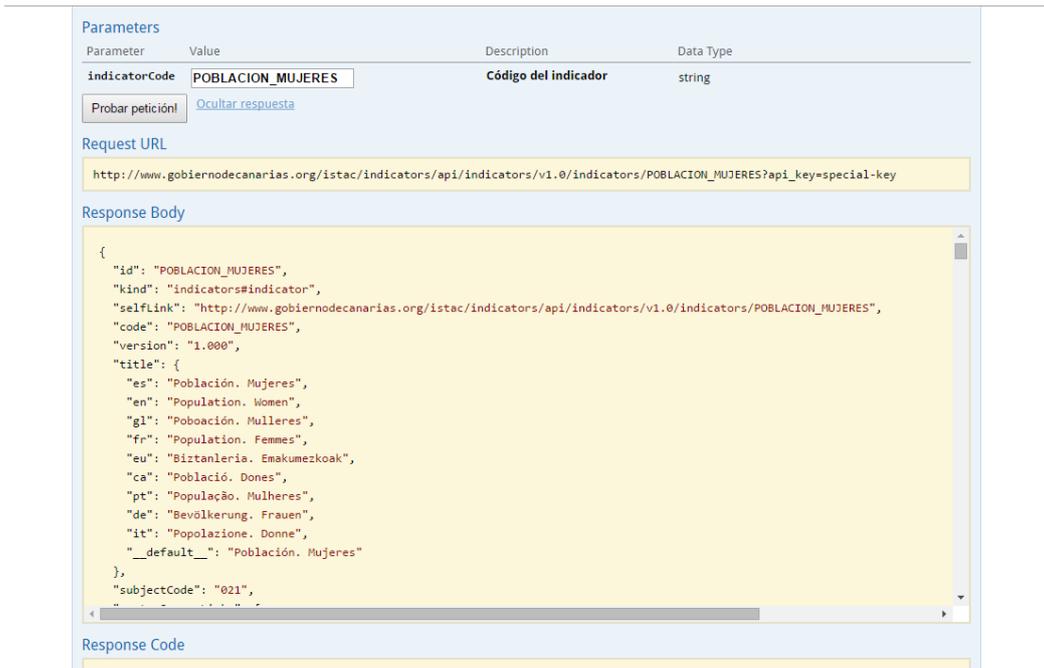
Por último, para las representaciones gráficas de los datos se utiliza la librería D3.js, la cual se trata más en profundidad en el apartado 4.5.2 Librerías (de visualización de datos).

2.1.3 API ISTAC

En cuanto a las herramientas disponibles para la obtención de los datos destaca la **API** [10] que provee el ISTAC, a la cual se realizan las peticiones, siendo devueltos en formato **JSON** [11]. Por otro lado, sí que dispone de documentación, muy útil para que el desarrollador pueda conocer tanto cómo realizar las consultas, como de qué manera se disponen los datos recibidos, qué datos se pueden obtener actualmente, qué describen los datos seleccionados, en qué unidades se ofrecerán los datos, etc.

Actualmente, la API cuenta con información sobre indicadores (véase apartado 3.3.), sistemas de indicadores (como por ejemplo, el listado de indicadores del SIMAC, véase apartado 3.2.), categorías de los indicadores y un sistema de referencia geográfica que permite clasificar las distintas organizaciones territoriales que existen en Canarias en sus diversos niveles (regional, insular, comarcal y municipal).

La documentación hace uso de la herramienta **Swagger** [12], *framework* para la creación de interfaces para APIs. Gracias a esta herramienta, resulta mucho más intuitivo navegar a través de las diversas rutas de la API, así como conocer cómo se generan los enlaces de consulta en función de los parámetros que se quieran incluir y además permite realizar pruebas sobre peticiones al instante.



The screenshot displays the Swagger tool interface for a REST API. It is divided into several sections:

- Parameters:** A table with columns 'Parameter', 'Value', 'Description', and 'Data Type'. The parameter 'índicatorCode' has the value 'POBLACION_MUJERES', description 'Código del indicador', and data type 'string'. Below the table are buttons for 'Probar petición!' and 'Ocultar respuesta'.
- Request URL:** A text box containing the URL: `http://www.gobiernodecanarias.org/istac/indicadores/api/indicadores/v1.0/indicadores/POBLACION_MUJERES?api_key=special-key`.
- Response Body:** A large text area showing the JSON response. The response is a JSON object with the following structure:

```
{
  "id": "POBLACION_MUJERES",
  "kind": "indicators#indicator",
  "selfLink": "http://www.gobiernodecanarias.org/istac/indicadores/api/indicadores/v1.0/indicadores/POBLACION_MUJERES",
  "code": "POBLACION_MUJERES",
  "version": "1.000",
  "title": {
    "es": "Población. Mujeres",
    "en": "Population. Women",
    "gl": "Poboación. Mulleres",
    "fr": "Population. Femmes",
    "eu": "Biztanleria. Emakumezkoak",
    "ca": "Població. Dones",
    "pt": "População. Mulheres",
    "de": "Bevölkerung. Frauen",
    "it": "Popolazione. Donne",
    "_default_": "Población. Mujeres"
  },
  "subjectCode": "021",
  ...
}
```
- Response Code:** A section at the bottom, partially visible, showing the status code of the response.

Figura 2.1 Ejemplo de petición y respuesta en la herramienta Swagger

2.2 Antecedentes software

Existen diversas herramientas similares a la que se desea desarrollar en este proyecto. Con ellas se pueden visualizar datos estadísticos sobre diversas áreas y en diversos entornos, por lo que puede resultar interesante realizar un análisis previo con el fin de que la aplicación objetivo ofrezca las mismas posibilidades y siga ciertos estándares de usabilidad que ya han establecido estas otras herramientas.

El antecedente software más directo, ligado a la tecnología móvil, es PULSEC [13], un proyecto convertido en trabajo de fin de grado de un antiguo alumno de la Universidad de La Laguna. Se trata de una aplicación para Firefox OS en la cual se pueden consultar los datos de indicadores del ISTAC, y analizar su representación numérica y gráfica. PULSEC, a diferencia de la aplicación objetivo de este proyecto, descargaba primero la base de datos del indicador que eligiese el usuario, y posteriormente se permitía filtrarlos en función de los criterios elegidos por el usuario, dado que durante la realización de este proyecto no estaba aún disponible la API que actualmente ofrece el ISTAC, cuyas peticiones ya indican la selección concreta de los datos requeridos. En esta aplicación, debe seleccionarse el indicador a consultar, tras la descarga de los datos del mismo, el filtrado que se especifique gracias a las opciones disponibles en un menú lateral, se reflejarán automáticamente en la visualización de los datos. Dispone de una caja de botones, gracias a los cuales se puede variar entre el tipo de representación gráfica que se puede visualizar en cada momento. Para cambiar de indicador, se vuelve hacia atrás y se repite el proceso.



Figura 2.2 Pantalla de inicio de PULSEC

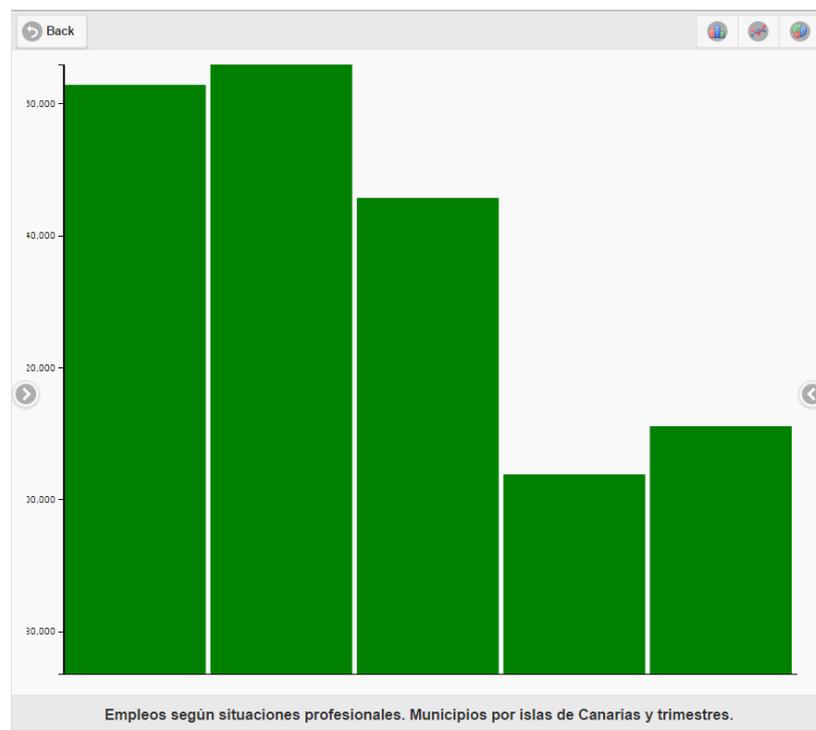


Figura 2.3 Resultados de una consulta en PULSEC

Otra de estas herramientas es la página web de consulta de indicadores económicos del EUROSTAT [14], la cual permite tras seleccionar un indicador en concreto, indicar el país para el cual se quieren consultar los datos, y posteriormente, también se puede seleccionar el criterio de representación de

Parte II.

Cuerpo principal. Descripción del trabajo

Capítulo 3.

Indicadores económicos

3.1 Introducción

Se denominan indicadores económicos a una serie de datos estadísticos recogidos en relación con algún tema concreto de interés analítico. Gracias a estos datos podemos conocer su variación en el tiempo, así como la tendencia que pueden seguir en el futuro estos datos en relación con sucesos actuales o incluso también en relación con algún despunte que se haya observado en algún otro indicador.

En cuanto a los indicadores ambientales, constituyen un instrumento útil para análisis y difusión de la mejor información disponible, ayudando a descubrir fenómenos complejos y sintetizando una gran cantidad de datos que, de otro modo, sería muy difícil ofrecer a la opinión pública.

Es de suma importancia el análisis de estos indicadores para poder predecir cambios importantes tanto en el entorno natural como económico; conociendo la tendencia de los datos, se debe analizar este comportamiento de los mismos, qué puede ocurrir si las cifras continúan en la misma línea durante un tiempo determinado y en caso de que esto sea desfavorable, qué medidas pueden tomarse para evitar pérdidas y desperfectos a tiempo.

3.2 SIMAC

Los indicadores ambientales son sistemas de medición del estado del medio ambiente, que permiten a la sociedad determinar el estado y evolución de una situación o un fenómeno en particular (por ejemplo, gases de efecto invernadero en la atmósfera o nitratos en el agua).

Gracias a los indicadores listados en la página web del Sistema de Indicadores Medio Ambientales de Canarias (SIMAC) se puede obtener una referencia sobre los datos que resultan interesantes para esta área.

El origen de SIMAC está en el Proyecto Interreg III-B INDICE [16], en el que se marcó como objetivo el análisis del sistema de indicadores medioambientales propuesto por la UE y la viabilidad de cálculo al menos para los ámbitos regionales e insulares de la Macaronesia; y, por otro lado, la definición de las medidas a adoptar para posibilitar su implantación y actualización regular en los respectivos ámbitos territoriales y la propuesta, en su caso, de un sistema de indicadores complementarios que reflejen la problemática específica de la Región Macaronésica.

El conjunto de indicadores seleccionados constituye un sistema de indicadores ambientales y de sostenibilidad común a la Región Macaronésica (Azores, Madeira, Canarias, y Cabo Verde), que permite evaluar comparativamente los efectos de las distintas políticas ambientales y la evolución de la región hacia un desarrollo sostenible.

3.3 Selección de indicadores

En este proyecto se han seleccionado una serie de indicadores estadísticos, recogidos por el Instituto Canario de Estadística, y que han resultado de gran interés conocer al área de Medioambiente y Sostenibilidad del Gobierno de Canarias. Se listan a continuación los indicadores seleccionados:

Selección de indicadores					
Indicador	Tipo	Descripción	Factor geográfico	Factor temporal	Unidades
Superficie	Básico	Espacio terrestre y marítimo correspondiente a cada territorio	Islas	Años	Kilómetros cuadrados
Población	Básico	Número de personas empadronadas en una circunscripción concreta	Islas	Años	Personas
Población. Hombres	Básico	Número de varones empadronados en una circunscripción concreta	Islas	Años	Personas
Población. Mujeres	Básico	Número de mujeres empadronadas en una circunscripción concreta	Islas	Años	Personas
Población. De 0 a 14 años	Básico	Número de personas cuya edad se encuentra entre los 0 y los 14 años de edad (ambos inclusive) en una circunscripción concreta	Islas	Años	Personas

Población. De 15 a 64 años	Básico	Número de personas cuya edad se encuentra entre los 15 y los 64 años de edad (ambos inclusive) en una circunscripción concreta	Islas	Años	Personas
Población. A partir de 65 años	Básico	Número de personas cuya edad se encuentra de los 65 años de edad en adelante en una circunscripción concreta.	Islas	Años	Personas
Población. Hombres de 0 a 14 años	Básico	Número de varones cuya edad se encuentra entre los 0 y los 14 años de edad (ambos inclusive) en una circunscripción concreta	Islas	Años	Personas
Población. Hombres de 15 a 64 años	Básico	Número de varones cuya edad se encuentra entre los 15 y los 64 años de edad (ambos inclusive) en una circunscripción concreta	Islas	Años	Personas
Población. Hombres a partir de 65 años	Básico	Número de varones cuya edad se encuentra de los 65 años de edad en adelante en una circunscripción concreta	Islas	Años	Personas
Población. Mujeres de 0 a 14 años	Básico	Número de mujeres cuya edad se encuentra entre los 0 y los 14 años de edad (ambos inclusive) en una circunscripción concreta	Islas	Años	Personas
Población. Mujeres de 15 a 64 años	Básico	Número de varones cuya edad se encuentra entre los 15 y los 64 años de edad (ambos inclusive) en una circunscripción concreta	Islas	Años	Personas
Población. Mujeres a partir de 65 años	Básico	Número de mujeres cuya edad se encuentra de los 65 años de edad en adelante en una circunscripción concreta	Islas	Años	Personas
Densidad de población	Derivado	Concentración de población residente en la superficie de un territorio en concreto	Islas	Años	Personas/Km ²
Densidad de población (más población turística)	Derivado	Ídem anterior, incluyendo también la población turística residente en dicho territorio	Islas	Años	Personas/Km ²
Razón de masculinidad	Derivado	Tasa de varones con respecto al número de mujeres en un territorio concreto	Islas	Años	
Índice de envejecimiento	Derivado	Porcentaje de personas ancianas con respecto al total de la población de un territorio	Islas	Años	Porcentaje

Índice de dependencia	Derivado	Establece la relación entre el grupo de la población potencialmente activa y los grupos de individuos económicamente dependientes	Islas	Años	
Ganado Bovino	Básico	Número de cabezas de ganado bovino registradas en un territorio concreto	Islas	Años	Cabezas
Ganado Ovino	Básico	Número de cabezas de ganado ovino registradas en un territorio concreto	Islas	Años	Cabezas
Ganado Caprino	Básico	Número de cabezas de ganado caprino registradas en un territorio concreto	Islas	Años	Cabezas
Ganado Porcino	Básico	Número de cabezas de ganado porcino registradas en un territorio concreto	Islas	Años	Cabezas
Concentración de ganado por superficie	Derivado	Concentración de ganado registrado en la superficie de un territorio en concreto	Islas	Años	Cabezas/Km ²
Plazas ofertadas por alojamientos turísticos	Básico	Número de plazas ofertadas por establecimientos de alojamiento turístico en un territorio	Islas	Años	Plazas
Índice censal de ocupación de plazas	Básico	Relación expresada en porcentaje entre el total de las pernoctaciones y el número de plazas ofertadas por establecimientos de alojamiento turístico en un territorio	Islas	Años	Porcentaje
Viajeros entrados en alojamientos turísticos	Básico	Número de viajeros registrados en establecimientos de alojamiento turístico en un territorio	Islas	Años	Personas
Pernoctaciones en alojamientos turísticos	Básico	Número de pernoctaciones registradas en establecimientos de alojamiento turístico en un territorio	Islas	Años	Noches
Población turística equivalente en alojamientos turísticos	Básico	Número de turistas registrados en establecimientos de alojamiento turístico en un territorio	Islas	Años	Personas

Plazas ofertadas por alojamientos turísticos por superficie	Derivado	Relación entre el número de plazas de alojamiento turístico disponibles en un territorio y su superficie	Islas	Años	Plazas/Km ²
Plazas ofertadas por alojamientos turísticos por cada 1.000 habitantes	Derivado	Número de plazas de alojamiento turístico ofertadas por cada mil habitantes de un territorio	Islas	Años	
Población turística equivalente en alojamientos turísticos por superficie	Derivado	Relación entre el número de turistas registrados en establecimientos de alojamiento turístico en un territorio y su superficie	Islas	Años	Personas/Km ²
Población turística equivalente en alojamientos turísticos por cada 1.000 habitantes	Derivado	Número de turistas registrados en establecimientos de alojamiento turístico por cada mil habitantes de un territorio	Islas	Años	
Parque de vehículos	Básico	Número de vehículos matriculados en un territorio	Islas	Años	Vehículos
Parque de vehículos. Turismos	Básico	Número de turismos matriculados en un territorio	Islas	Años	Vehículos
Índice de motorización	Derivado	Número de vehículos por cada mil habitantes de un territorio	Islas	Años	
Índice de motorización (más población turística)	Derivado	Número de vehículos por cada mil habitantes de un territorio, además de su población turística registrada	Islas	Años	
Índice de motorización (turismos)	Derivado	Número de turismos por cada mil habitantes de un territorio	Islas	Años	
Índice de motorización (más población turística) (turismos)	Derivado	Número de turismos por cada mil habitantes de un territorio, además de su población turística registrada	Islas	Años	
Concentración de vehículos por superficie	Derivado	Relación entre los vehículos matriculados en un territorio y su superficie	Islas	Años	Vehículos/Km ²
Concentración de turismos por superficie	Derivado	Relación entre los turismos matriculados en un territorio y su superficie	Islas	Años	Vehículos/Km ²

Tabla 3.1 Selección de indicadores

Capítulo 4.

Visualización de datos

4.1 Introducción a la visualización de datos

El objetivo de la visualización de datos es representar gráficamente una serie de datos estadísticos, para así facilitar su interpretación incluso a un público inexperto en el análisis estadístico. Dado que este tipo de representación aumenta notablemente su comprensión, también influye en la toma de decisiones referentes a los mismos, por lo que es vital que estas representaciones sean lo más claras y precisas posibles, y que no den la sensación de ser un conjunto de valores y hechos no relacionados entre sí.

4.2 Formato de los datos

Para poder realizar un trabajo de visualización de datos es importante partir de una buena base de información, en la que el formato de los datos recogidos debe ser de calidad, cumpliendo con unos mínimos de fiabilidad, precisión, relevancia, y además deben estar actualizados, bien identificados y estructurados correctamente.

Existen diversos estándares (algunos denominados como *lenguajes* por su uso extendido) para la representación de los datos, por medio de los cuales se consiguen los requisitos estructurales y de identificación que debe ofrecer un conjunto de datos para su tratamiento. Algunos de ellos son:

- **XML:** Extensible Markup Language [17]. Es un formato bastante popular en la web. Consiste en presentar la información etiquetada y con los correspondientes atributos que describen el dato que contienen. Las etiquetas pueden anidarse para dar lugar a estructuras complejas de datos.
- **CSV:** Comma Separated Values [18]. Se presentan los datos como si se trataran de tablas, separando los valores de una misma fila por medio de un carácter separador (normalmente, una coma), y separando cada

fila con un salto de línea. No es un formato adecuado para la visualización de datos, dado que no contiene ninguna descripción de los mismos; sin embargo, puede resultar muy útil para el tratamiento de datos (si se conoce su significado) por su sencillez para ser tratados por software.

- **JSON:** JavaScript Object Notation. Es actualmente uno de los formatos más extendidos que ofrecen las APIs web. Este lenguaje es bastante legible para los humanos y también fácilmente procesable para las máquinas, puesto que usa pares de clave-valor para representar los datos, de manera similar a como haría XML, pero simplificando el código ya que se evita el uso excesivo de etiquetas típico en XML. Un ejemplo de código en JSON sería el siguiente:

```
{
  "id": 1245,
  "title": {
    "es": "Título",
    "en": "Title"
  },
  "relatedWith": [427, 867, 1352]
}
```

En este proyecto, se trabaja con el formato JSON, ya que la API ISTAC los ofrece en este formato. La documentación proporcionada desde el ISTAC explica cómo obtener los datos, qué significan y de qué manera están estructurados en el formato JSON. Gracias a esta información, se pueden programar tanto las peticiones de información personalizada, como la posterior visualización de los datos.

4.3 Técnicas de visualización de datos

Dada la variedad de datos que pueden representarse gráficamente y las implicaciones que puede tener la unidad en la que se ofrecen los datos en lo que se pretende visualizar por medio de un gráfico, existe también una gran variedad de formas de técnicas de representación gráfica. A continuación se enumeran las más utilizadas:

4.3.1 Gráficos de barras

Se representan de manera simple los datos en forma de barras, cuya longitud varía en función del valor del dato. Las barras pueden presentarse en vertical u horizontal, y lo normal es que crezcan hacia arriba o hacia la derecha, respectivamente.

A partir de este modelo simple, derivan otros más complejos, como los gráficos de barras compuestas, donde cada barra se divide en componentes de ese dato, y se aprovecha para representar subdivisiones de esos componentes dentro de la propia barra. En los gráficos de barras comparativas, se muestra un conjunto de barras que permiten comparar varias series simultáneamente; entre otros.

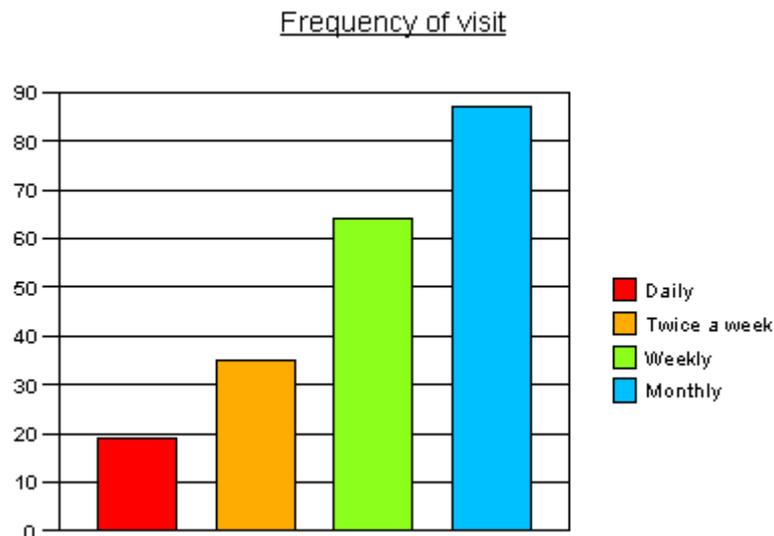


Figura 4.1 Ejemplo de gráfico de barras [Fuente: Snapsurveys]

Este tipo de gráficos es muy común en la representación de datos numéricos simples, ya sean discretos o continuos.

4.3.2 Gráficos de líneas

El gráfico de líneas es muy similar al gráfico de barras. En él, se unen los puntos correspondientes a los valores mediante segmentos, que formarán una línea quebrada o curva (si se suavizan las pendientes). Los valores de los datos se representan en el eje vertical, mientras que la variación de los mismos se refleja a lo largo del eje horizontal. Por tanto, la línea (o líneas, si se representan varias series que varían en rangos similares) *avanzará* hacia la derecha.

Se puede transformar un gráfico de barras en uno de líneas y viceversa de forma sencilla, dado que los puntos donde cambian las pendientes de las líneas se corresponden a la altura que debe tener la barra correspondiente a ese valor.

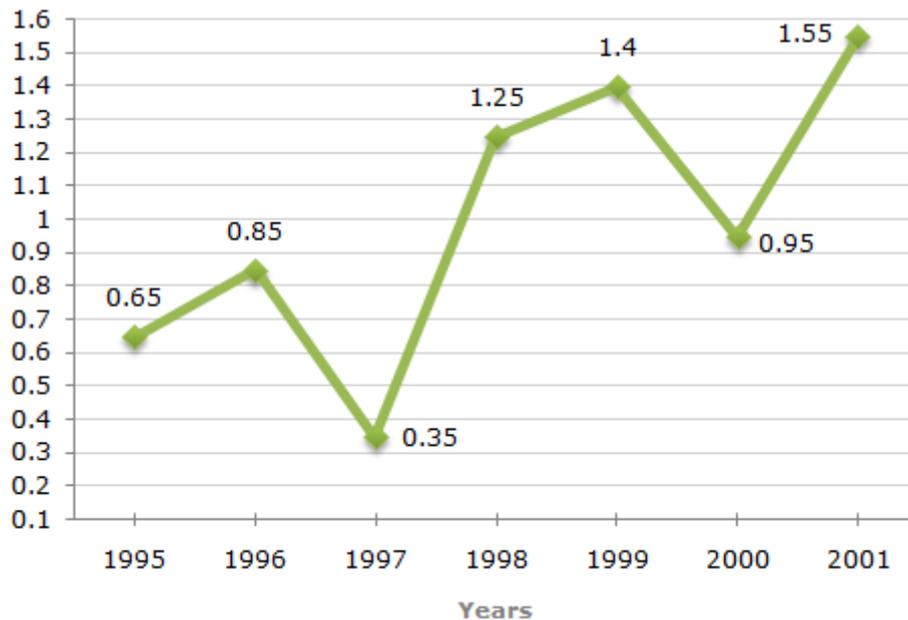


Figura 4.2 Ejemplo de gráfico de barras [Fuente: IndiaBIX]

Al igual que los gráficos de barras, los gráficos de líneas son útiles para representar datos numéricos discretos o continuos.

4.3.3 Gráficos circulares o de sectores

En estos gráficos se representan los valores de los datos en forma de sectores del área de un círculo, de manera que se puede visualizar el impacto de un valor con respecto al total de los datos.

Normalmente, cada sector en los que se divide el círculo está relleno de un color representativo, y los valores que se representan son porcentajes. De esta forma, completando el círculo se representa el 100% de los datos.

El círculo puede estar representado en dos o tres dimensiones, aunque este aspecto suele ser meramente estético.

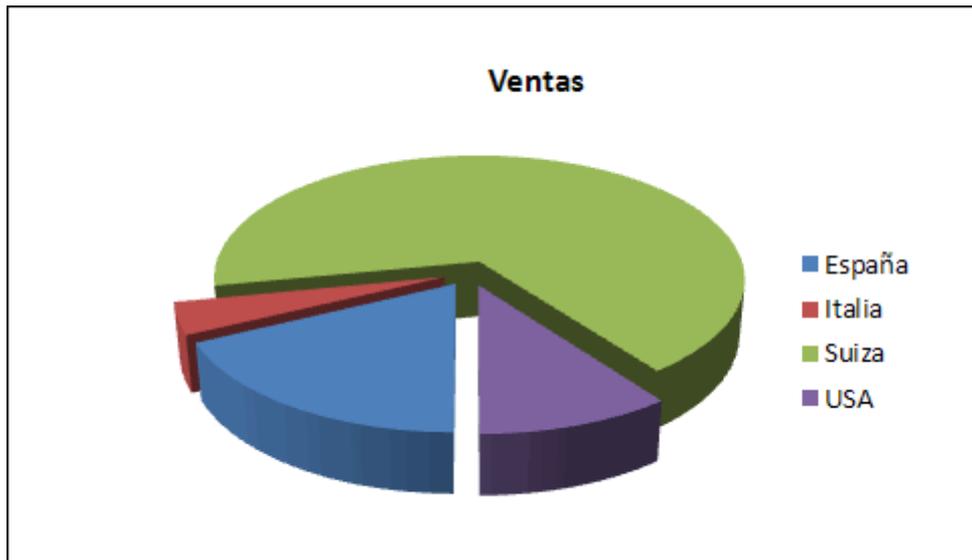


Figura 4.3 Ejemplo de gráfico circular o de sectores [Fuente: ADR Formación]

Las técnicas de visualización escogidas para este proyecto han sido los gráficos de línea y los gráficos de barra, dado que éstos son los que reflejan más claramente la variación en el tiempo que puede tomar el valor de un indicador y permiten hacer una mejor idea de la tendencia que puede tomar dicho valor.

Para la representación gráfica de los datos en este proyecto, se utilizará la librería D3.js, expuesta más a fondo en el apartado 4.5.2 Librerías (de visualización de datos).

4.4 Características deseables

En este proyecto, queda establecido que la información debe poder ser visualizada de manera numérica y gráfica, pero para que esta visualización pueda ser interpretada de la mejor manera posible, deben tenerse en cuenta ciertos aspectos que faciliten el análisis de los datos y que permitan al usuario localizar la información que busca dentro de un resultado que puede ser realmente extenso.

Por tanto, y en cuanto a la visualización **numérica** de los datos se han tenido en cuenta los siguientes aspectos:

- La agrupación de los datos se realiza primero por islas, y posteriormente por años.

- Se ha tenido muy en cuenta el formato de los rótulos, presentando los nombres de las islas en negrita y contenidos en unas cabeceras. A este titular, le siguen una serie de tarjetas con sangrías laterales que hacen ver que *pertenecen* a la isla nombrada con anterioridad. Dentro de estas tarjetas, se muestra a modo de cabecera el año en cuestión, e inmediatamente el dato referente a dicho año en dicha isla.
- Junto al dato, siempre se muestra la unidad en la que se ofrece el mismo.
- También se ofrece una comparativa (normalmente en porcentaje, y cuando no es posible, simplemente el dato total) del dato homólogo para el conjunto del archipiélago canario, de forma que se pueda observar de manera directa el impacto de ese dato en concreto en el total de Canarias.



Figura 4.4 Ejemplo de salida de los datos numéricos de una consulta

Por otro lado, los aspectos a tener en cuenta para la representación **gráfica** de los datos son:

- Las gráficas deben tener un área considerable para no perjudicar la representación de los datos.

- Al tratarse este proyecto de una aplicación para móviles, puede que algunos dispositivos dispongan de una pantalla de dimensiones reducidas y que las gráficas no puedan visualizarse con la claridad que aportaría tener una pantalla de mayores dimensiones. Para aliviar este problema, se ha de tener en cuenta la adaptación de las dimensiones de los gráficos ante un posible giro horizontal de la pantalla. En esta posición, las dimensiones de la gráfica mejoran notablemente la interpretación de la información.
- Los ejes deben tener anotaciones sobre la escala de valores, para así facilitar sobre qué cantidades oscila el dato observado.
- El eje vertical siempre representará los valores que puede tomar el dato, mientras que el eje horizontal representará la línea temporal, dividida en años.
- La escala de valores que representan los ejes debe siempre ajustarse al rango de los datos que se representarán en su interior. De esta forma, se permite apreciar de manera óptima la tendencia que siguen los datos.
- Como titular de la gráfica debe indicarse qué dato representa y la unidad en la que se miden los mismos.
- Los datos referentes para las islas se muestran en una única gráfica con diversas series correspondientes a las tendencias de cada isla.
- Dichas series se diferencian entre sí por el color. Para poder identificar cada color con una isla, se acompaña con una leyenda que indique el color perteneciente a cada isla.
- Volviendo una vez más a las limitaciones que podría tener una pantalla pequeña, esta leyenda debe poder mostrarse y ocultarse para que no resulte demasiado intrusiva.
- De manera análoga a como ocurre con la representación numérica, debe mostrarse la referencia al total del archipiélago. Para ello, se representa en otra gráfica el rango de datos que se representaba anteriormente para las islas, pero con los datos referentes a Canarias. La comparativa puede hacerse al visualizar ambas gráficas simultáneamente.



Figura 4.5 Ejemplo de salida de los datos gráficos para las islas de una consulta

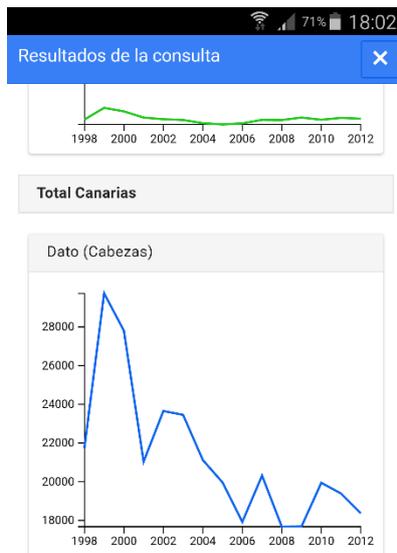


Figura 4.6 Ejemplo de salida de los datos gráficos para Canarias de una consulta

4.5 Servicios de visualización

Existen diversas herramientas que permiten hacer uso de la visualización de datos. A continuación se enumeran algunos ejemplos:

4.5.1 Aplicaciones

Hojas de cálculo

Es habitual encontrar alguna utilidad de hojas de cálculo en cualquier *suite* ofimática. En las más utilizadas, se encuentra la opción de generar gráficos a partir de los datos introducidos en dichas hojas. Algunos de estos programas son gratuitos, como LibreOffice Calc [19] u Hojas de Cálculo de Google [20]. Otros son de pago, como Microsoft Office Excel [21].

Tableau Public

Tableau Public [22] es una herramienta que se presenta en versión gratuita o de pago que permite crear mapas interactivos, gráficos de barras, circulares, etc. Permite la importación de tablas Excel, ficheros de texto e incluso servidores de bases de datos para facilitar el trabajo.

4.5.2 Librerías

D3.js

Data-Drive Documents (D3) [23] es una librería utilizada para visualización de datos a través de estándares web como HTML5, CSS3, JavaScript y gráficos vectoriales en formato SVG y consta de un amplio abanico de posibilidades a la hora de realizar representaciones gráficas y de ejemplos de uso y desarrollo de los mismos en su documentación web.

La creación de gráficas con D3 parte de los elementos más sencillos incorporados por el nuevo estándar HTML5, como puntos, líneas y demás figuras geométricas que pueden ser representadas en cualquier elemento *canvas*. Basta con indicar a la librería las dimensiones del lienzo donde se mostrará nuestra gráfica, el rango de los datos con el fin de ajustar los ejes en función de estos valores, indicar las series de datos que se quieren representar, y por último, fijar el estilo de cada uno de los elementos visuales de la gráfica. D3,

aparte de automatizar gran parte del proceso anteriormente descrito, también está provista de múltiples funciones de utilidad, como selectores de elementos DOM que facilita el manejo de dichos elementos, funciones útiles para el trabajo con *arrays*, dado que es la estructura de datos mediante la cual se insertan los datos en la gráfica, o funciones relacionadas con aspectos del estilo, como funciones de transformación del formato de colores, por ejemplo.

Las visualizaciones gráficas que genera pueden también añadir características interactivas y su perfecta adaptación a aplicaciones basadas en tecnologías web gracias al uso de estándares actuales y la gran cantidad de documentación y ejemplos que pueden encontrarse en su página web la hacen idónea para su uso en este proyecto.

Se adjunta un apéndice (Apéndice A – Creación de gráficas con D3) a esta memoria en la que se muestra un extracto escrito en JavaScript con el cual se puede crear un gráfico de líneas usando D3.js.

Google Developers Charts Tools

La herramienta de Google de gráficos para desarrolladores [24] permite generar gráficos tanto en HTML5 como en SVG. Es totalmente gratuita y dispone de diversos tipos de gráficas personalizables.

Las gráficas pueden ser interactivas, y de cara al desarrollador es bastante sencillo tanto crear las gráficas y asociar los datos que contiene como personalizar los diversos aspectos de diseño de las mismas.

Capítulo 5.

Desarrollo

5.1 Plan de trabajo

En la siguiente tabla se muestra la distribución del plan de trabajo seguido en el desarrollo de este proyecto.

Distribución temporal de tareas	
Semanas 1-2	Actividad 2. Revisión de antecedentes y estado del arte.
Semanas 3-5	Consulta de información a la API ISTAC: búsqueda de información, formación, desarrollo y pruebas. Seminario 1.
Semanas 4-8	Visualización de información: búsqueda de información, formación, desarrollo y pruebas. Seminario 2.
Semanas 9-11	Desarrollo del prototipo para dispositivos móviles Android: búsqueda de información, formación, desarrollo y pruebas. Seminario 3 y comienzo de redacción de la memoria.
Semanas 12-14	Desarrollo del prototipo para dispositivos móviles iOS: búsqueda de información, formación, desarrollo y pruebas. Seminario 4 y redacción de la memoria.
Semana 15	Pruebas. Redacción de la memoria.
Semana 16	Actividad 8. Preparación de la presentación

Tabla 5.1 Distribución temporal del plan de trabajo

5.2 Tecnologías empleadas

En líneas generales, gran parte de la aplicación se basa en la tecnología AJAX [25] para la petición asíncrona de datos. Gracias al uso de llamadas AJAX, no es necesario recargar una página para obtener los datos.

A continuación, se aporta un esquema sobre el flujo de peticiones AJAX de la aplicación y la utilización de los datos recibidos:

5.2.1 Petición 1: Inicialización de la aplicación

Al iniciar la aplicación, ésta debe conectarse al servidor *banot*, alojado en la ESIT de la Universidad de La Laguna donde se han desplegado dos ficheros en formato JSON:

- Uno de los ficheros contiene un listado de *indicadores permitidos* que son los que se mostrarán en la aplicación. El motivo de la creación de este fichero es que en la API que ofrece el ISTAC existen indicadores que no resultan relevantes para el área de Medioambiente y Sostenibilidad, por lo que gracias a este archivo se puede hacer una selección de los indicadores que (previo análisis y confirmación por parte de personal del ISTAC) sí resultan interesantes de observar.
- El otro fichero contiene la información de los *indicadores derivados*, y de él se obtienen las características necesarias para conocer de qué indicadores básicos precisa, y cuál es la fórmula para calcular los datos de este indicador, además de cuestiones más básicas como su título en castellano o su código.



Figura 5.1 Diagrama de Petición 1: Inicialización

5.2.2 Petición 2: Categorías

Una vez se han inicializado los primeros datos internos de la aplicación, debe obtenerse el listado de categorías:

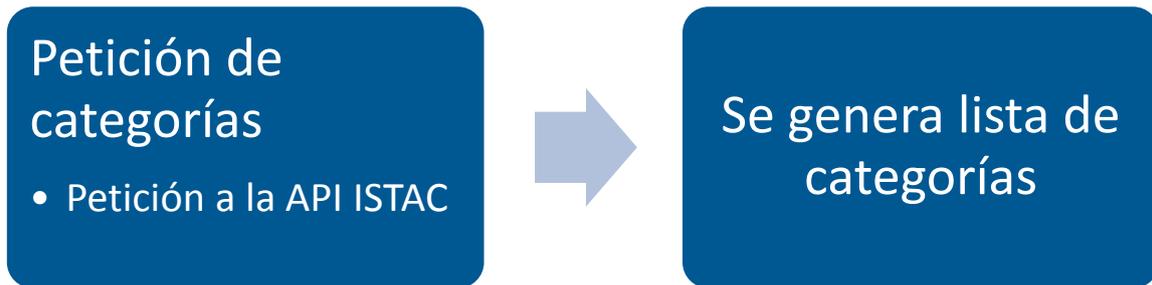


Figura 5.2 Diagrama de Petición 2: Categorías

5.2.3 Petición 3: Indicadores de la categoría seleccionada

Tras haber seleccionado una categoría, se mostrarán los indicadores (tanto básicos como derivados) pertenecientes a dicha categoría:

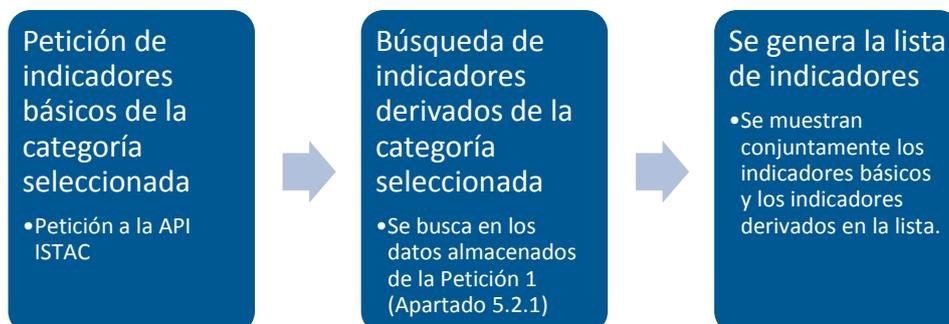


Figura 5.3 Diagrama de Petición 3: Indicadores

5.2.4 Petición 4: Datos del indicador para el filtro de consulta

Tras escoger un indicador, se mostrarán los posibles años e islas disponibles de los datos en un formulario, mediante el cual se puede filtrar la consulta. En este apartado el proceso difiere, en función de si el indicador es básico o derivado.

Si el indicador es básico:

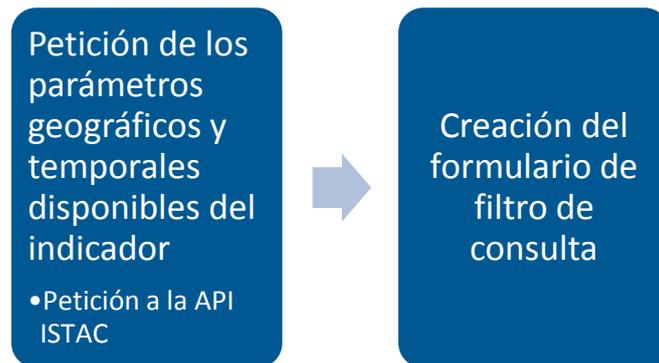


Figura 5.4 Diagrama de Petición 4: Parámetros geográficos y temporales de un indicador básico

Si el indicador es derivado, se debe tener en cuenta que sólo se podrá hacer cálculos posteriormente con los datos si van referidos a la misma isla y en el mismo año, por lo que los datos descargados deben compartir estos parámetros:

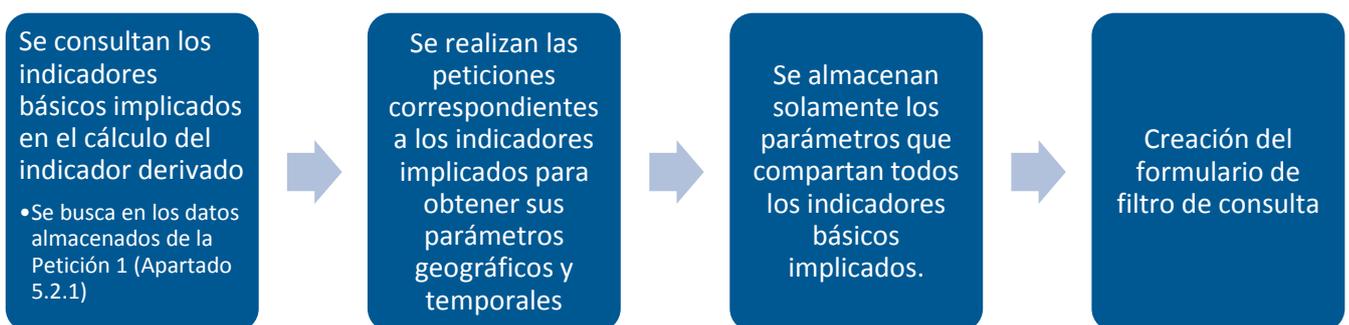


Figura 5.5 Diagrama de Petición 4: Parámetros geográficos y temporales de un indicador derivado

5.2.5 Petición 5: Datos estadísticos del indicador

Una vez que el usuario ejecuta la consulta, se deben rellenar las vistas de visualización numérica y gráfica con los datos recibidos. Nuevamente, existen diferencias en el proceso en función del tipo de indicador que haya seleccionado el usuario.

Si se trata de un indicador básico:

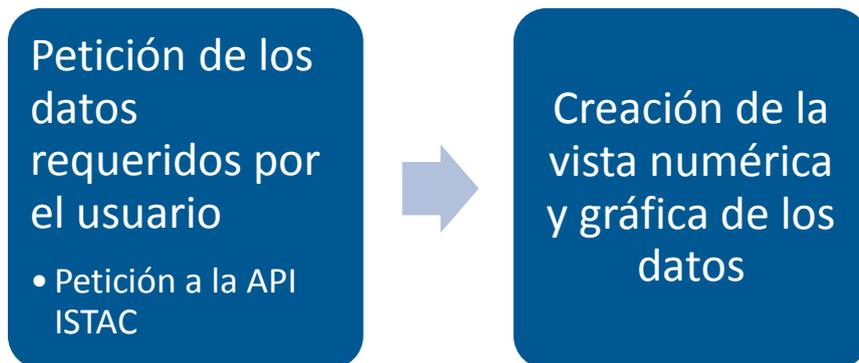


Figura 5.6 Diagrama de Petición 5: Datos estadísticos de un indicador básico

Si se trata de un indicador derivado, se debe además de realizar las peticiones de los datos, calcular el nuevo dato:

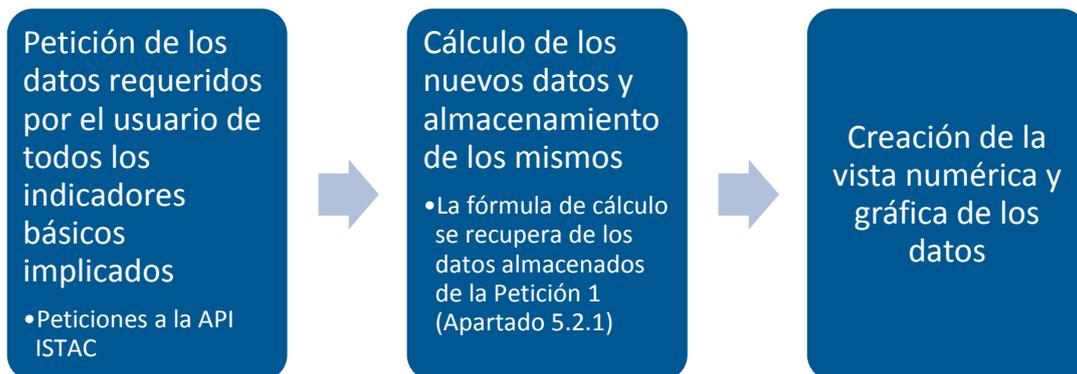


Figura 5.7 Diagrama de Petición 5: Datos estadísticos de un indicador derivado

5.3 Herramientas

Las herramientas que se han utilizado para desarrollar el proyecto son:

- **WebStorm [26]**: IDE de la compañía JetBrains (IntelliJ IDEA, Android Studio) para el desarrollo de aplicaciones web. Se ha escogido este IDE por sus múltiples funcionalidades, y la capacidad de incorporación de plugins que multiplican las capacidades de este programa y aportan un plus a la productividad en el desarrollo del proyecto.
- **Emuladores Android [27] e iOS [28]**: Gracias a estos emuladores se han hecho pruebas (en el caso de Android, también en dispositivos físicos) y se ha podido comprobar el correcto funcionamiento de la aplicación.
- **Git [29] y Github [30]**: Estas herramientas han permitido el uso de control de versiones y el alojamiento online del repositorio, respectivamente.
- **Trello [31]**: Herramienta de productividad online que permite la disposición de tareas en forma de tarjetas que se agrupan en diversas listas. Gracias a esta herramienta se han marcado tantos los hitos de desarrollo del proyecto como las funcionalidades pendientes de implementar. A continuación, se muestra el estado de listas y sus tarjetas, insertadas en las listas en orden cronológico inverso, a fecha de entrega de esta memoria:

Estado final de las listas en Trello	
Tareas pendientes	Tareas completadas
Cambiar icono y <i>splashscreen</i> de la aplicación	Añadir al fichero JSON de los indicadores derivados la unidad del indicador
Modificar página de “Acerca de” y añadir créditos de la ULL y la ESIT	Crear una gráfica multiserie, en lugar de múltiples gráficas de una serie cada una
	Obtener datos de indicadores derivados
	Crear fichero JSON de indicadores derivados
	Incorporar nuevas dimensiones temporales y geográficas limitadas a los indicadores medioambientales
	Seleccionar categorías de medio ambiente de la API
	Crear gráficas de los datos

	Crear formulario de filtros de consulta
	Incorporar D3.js al proyecto

Tabla 5.2 Estado final de las listas en Trello

5.4 Tratamiento de la información

Existen diversos ficheros JSON que se obtienen de la API del ISTAC, en función de los datos que se están consultando.

En primera instancia, se consultan las categorías de indicadores disponibles, y los campos del JSON devuelto que se utilizan en la aplicación son:

- “code”: Código de la categoría. Se usa dentro de la aplicación como el campo identificador de cada categoría.
- “title”: Nombre de cada categoría. Exactamente, se utiliza el subcampo referente al idioma español (es).

Luego, una vez se elige la categoría se realiza una petición a la API para obtener los indicadores pertenecientes a dicha categoría y que están permitidos por el otro fichero JSON donde se controla qué indicadores se muestran.

El JSON que contiene los indicadores permitidos está alojado en el servidor *banot* de la ESIT, y contiene, para cada indicador, los siguientes campos:

- “code”: Código del indicador. Se usa dentro de la aplicación como el campo identificador de cada indicador.
- “compare”: Esta marca permite saber si en la vista de resultados el dato del indicador en cuestión debe compararse en porcentaje con el total de canarias (si su valor es “percent”) o simplemente con el dato correspondiente a ese año en Canarias (si su valor es “total”).

El JSON que devuelve la API del ISTAC en referencia a los indicadores disponibles contiene los siguientes campos utilizados en la aplicación:

- “code”: Código del indicador. Se usa dentro de la aplicación como el campo identificador de cada indicador.
- “title”: Nombre del indicador. Exactamente, se utiliza el subcampo referente al idioma español (es).

- “subjectCode”: Código de la categoría a la que pertenece el indicador. Gracias a este campo se clasifican los indicadores según su categoría.

En el fichero JSON que describe los indicadores derivados (también alojado en *banot*), se ha tratado de seguir una estructura de datos similar a la que ofrecen los ficheros JSON proporcionados por la API. Para cada uno los indicadores en este fichero, se encuentran los siguientes campos:

- “code”: Código del indicador. Se usa dentro de la aplicación como el campo identificador de cada indicador.
- “title”: Nombre del indicador. Se ha omitido el subcampo del idioma y sólo se ha dado soporte al nombre en castellano.
- “indicators”: Conjunto de indicadores con los que se debe operar para obtener el resultado del indicador derivado. Consiste en un array de códigos de indicadores.
- “calculate”: Función gracias a la cual se calcula el valor del indicador derivado a partir de datos de indicadores que se pasan como parámetros. La función se define en forma de *string*, y posteriormente en el programa se hace uso de la función nativa de JavaScript *eval* para asignar esta función a una variable, interpretando la *string* como código. En cuanto a los parámetros, éstos se pasan a la función dentro de un *array*, dado que dependiendo de la función, puede requerirse un número diferente de parámetros.

Una vez se ha seleccionado un indicador, se debe realizar una nueva petición a la API para obtener los datos referentes a las fechas y lugares que se pueden consultar. Para ello, se utilizan estos campos del JSON devuelto:

- “code”: Código del indicador. Se usa dentro de la aplicación como el campo identificador de cada indicador.
- “title”: Nombre del indicador. Exactamente, se utiliza el subcampo referente al idioma español (es).
- “subjectCode”: Código de la categoría a la que pertenece el indicador. Gracias a este campo se clasifican los indicadores según su categoría.
- “dimension”: En este campo se diferencian tres dimensiones: geográfica, temporal y tipo de medidas. Dentro de cada una de las citadas

dimensiones, se encuentran los campos “granularity”, que hace referencia a las diferentes granularidades que puede ofrecer el marcador para cada una de las dimensiones; y también el campo “representation”, que contiene los valores geográficos, temporales y de medida disponibles para el indicador, acompañados de un atributo llamado “granularityCode”, que indica a qué tipo de granularidad pertenece el valor en cuestión. Se usan para la aplicación los siguientes campos de cada dimensión:

- “GEOGRAPHICAL”: Dentro de este campo, se utilizan del subcampo “representation” los datos referentes a los lugares cuya granularidad es de nivel insular (“granularityCode” = “ISLANDS”) y el dato de granularidad regional (“granularityCode = “REGIONS”) para poder obtener el dato referente a Canarias. Dentro de cada lugar, se usan los campos:
 - “code”: Código de la isla o región.
 - “title”: Nombre de la isla o región. Exactamente, se utiliza el subcampo referente al idioma español (es).
 - “granularityCode”: Código que permite clasificar cada lugar según su granularidad.
- “TIME”: Dentro de este campo, se utilizan del subcampo “representation” los datos referente a las fechas cuya granularidad es anual (“granularityCode = “YEARLY”). Dentro de cada fecha, se usan los campos:
 - “code”: Código de la fecha.
 - “title”: Nombre de fecha. Exactamente, se utiliza el subcampo referente al idioma español (es).
 - “granularityCode”: Código que permite clasificar cada fecha según su granularidad.
- “MEASURE”: Dentro de este campo, se utiliza del subcampo “representation” los datos referentes a las medidas absolutas de los datos (“code” = “ABSOLUTE”). Dentro de esta medida, se utilizan los campos:

- “code”: Código de la medida.
- “quantity.unit”: Nombre de la medida. Exactamente, se utiliza el subcampo referente al idioma español (es).

Por último, una vez se realiza una petición de datos de un indicador, los campos utilizados por la aplicación de los datos devueltos por la API son:

- “dimension”: En este campo se diferencian tres dimensiones:
 - “GEOGRAPHICAL”: Dentro de este campo, se usan datos del subcampo “representation.index”, que contiene pares clave-valor, en los que las claves son códigos de lugares, y los valores el índice del *array* de datos donde comienzan a representarse los datos referentes a dicho lugar.
 - “TIME”: Dentro de este campo, se usan datos del subcampo “representation.index”, que contiene pares clave-valor, en los que las claves son códigos de fechas, y los valores el índice del *array* de datos donde comienzan a representarse los datos referentes a dicha fecha.
 - “MEASURE”: Dentro de este campo, se usan datos del subcampo “representation.index”, que contiene pares clave-valor, en los que las claves son códigos de medidas, y los valores el índice del *array* de datos donde comienzan a representarse los datos referentes a dicha medida.
- “observation”: Es el *array* que contiene los datos que se piden en la consulta. Gracias al sistema de indexado anteriormente descrito y aplicando una fórmula de búsqueda se puede acceder a un dato en concreto conociendo lugar, fecha y medida del mismo. La fórmula de búsqueda es:

$$\text{Índice}_{dato} = (\text{Índice}_{lugar} * \text{Fechas.size} * \text{Medidas.size}) + (\text{Índice}_{fecha} * \text{Medidas.size}) + \text{Índice}_{medida}$$

Donde:

- Índice_{dato} es el índice del dato concreto que estamos buscando.
- Índice_{lugar} es el índice del lugar al que pertenece el dato.
- Índice_{fecha} es el índice de la fecha a la que pertenece el dato.

- $\text{Índice}_{\text{medida}}$ es el índice de la medida en la que deseamos obtener el dato.
- Fechas.size es el número de fechas que se han incluido en la consulta.
- Medidas.size es el número de medidas que se han incluido en la consulta.

5.5 Funcionamiento

Nada más iniciar la aplicación, se muestra la lista de categorías de indicadores disponibles en la plataforma.



Figura 5.8 Pantalla de categorías

Al seleccionar una de las categorías, se muestra la lista de indicadores que pertenecen a la categoría seleccionada.



Figura 5.9 Indicadores de la categoría “Población”

Una vez se elige un indicador, se muestran las islas y los años disponibles para este indicador.

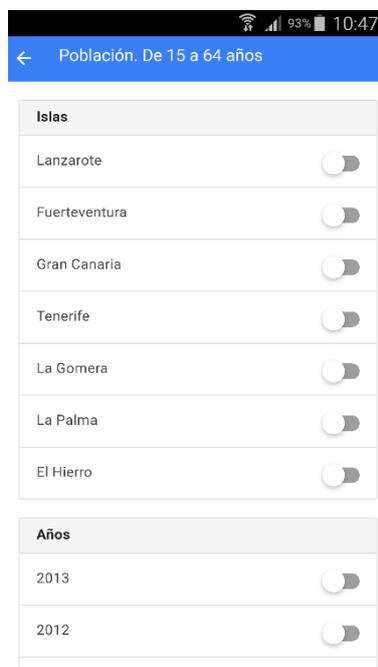


Figura 5.10 Islas y años disponibles para el indicador “Población. De 15 a 64 años”

En esta pantalla, el usuario puede elegir si filtra su consulta, seleccionando algunas islas y años. En caso de que no se seleccione ninguna de las opciones, se interpreta que se deben consultar todos los datos disponibles.

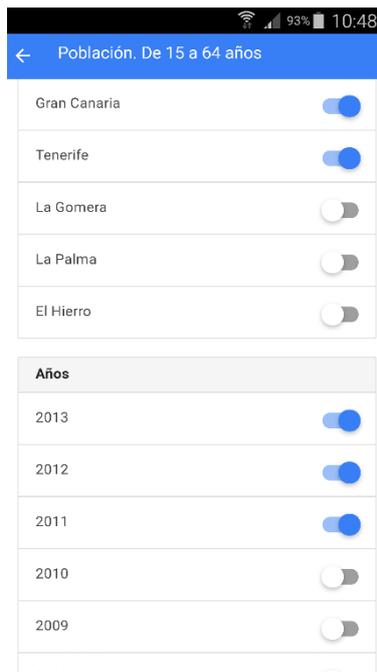


Figura 5.11 Selección de criterios de búsqueda

Una vez que el usuario personaliza los parámetros de su consulta, debe ejecutarla pulsando en el botón “Realizar consulta” que se encuentra al final de la pantalla de selección de parámetros de consulta.

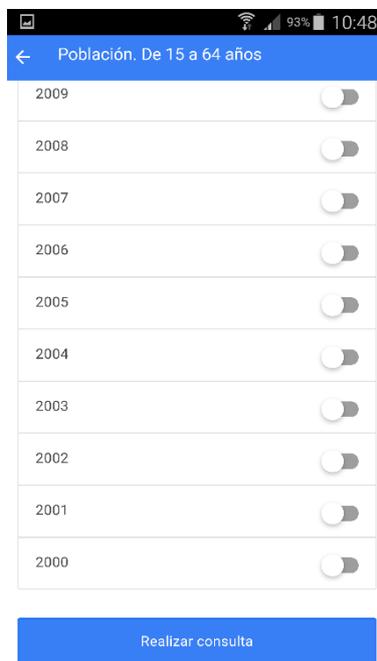


Figura 5.12 Botón “Realizar consulta”

Una vez se han recuperado los datos, éstos se muestran numéricamente por defecto, y ordenados primero por islas y después por años.



Figura 5.13 Resultados numéricos de la consulta

Si se desea visualizar la representación gráfica de los datos, se debe cambiar de pestaña, pulsando el botón con el icono de gráfica de sectores.

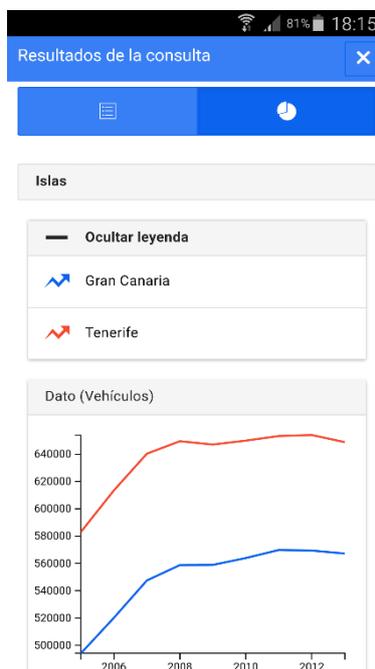


Figura 5.14 Resultados gráficos de la consulta

Como se había comentado con anterioridad, la leyenda se puede ocultar y mostrar, según le interese al usuario.

Para cerrar los resultados de la consulta, basta con pulsar el botón con forma de aspa de la esquina derecha superior de la pantalla. Para cambiar los parámetros de la consulta o consultar otro indicador, basta con navegar hacia atrás por las distintas pantallas usando el botón de volver atrás, situado en la esquina izquierda superior de la pantalla. En dispositivos Android, también se puede navegar hacia atrás pulsando el botón Atrás del dispositivo.

Para cerrar la aplicación:

- En iOS: Basta con pulsar el botón Home del dispositivo.
- En Android: Debe navegarse atrás hasta la pantalla de selección de categorías y pulsar nuevamente el botón Atrás del dispositivo. Se mostrará un mensaje de confirmación para salir de la aplicación.

5.6 Problemas encontrados

5.6.1 Cross-Domain

Cross-domain [32] es el problema que surge a la hora de realizar peticiones XMLHttpRequest (solicitudes HTTP/HTTPS mediante JavaScript para intercambiar información en cualquier formato de texto plano) a un servidor web distinto al que reside la aplicación y, por motivos de seguridad, los navegadores están diseñados para bloquear este tipo de peticiones.

Para solucionar este problema surge como primera aproximación JSONp (JSON with Padding) [33], una implementación del lado del servidor cuya finalidad es inyectar un script en la página actual de manera que sí permita acceder a los recursos situados en otros dominios.

Las peticiones AJAX realizadas a la API del ISTAC incluyen un parámetro que hacen referencia a JSONp, con la finalidad de que los datos puedan ser recuperados mediante esta implementación y evitando así el problema de Cross-Domain. Sin embargo, para las peticiones de los ficheros JSON alojados en el servidor de la ESIT, si trabajamos en un navegador web, nos encontramos con el citado problema. Afortunadamente, a la hora de ejecutar el proyecto como aplicación en un dispositivo o emulador de Android y en un emulador de iOS, no se ha observado este problema, y se pueden obtener correctamente los datos, dado que cross-domain es una política que siguen los navegadores web. Sin

embargo, para casos de necesidad, existen también extensiones [34] para navegadores como Google Chrome o Mozilla Firefox que permiten saltarse esta política cuando están activadas, asumiendo que el usuario es totalmente responsable de la ejecución del script, incluso si se trata de código malintencionado.

Capítulo 6.

Pruebas de usabilidad

6.1 Motivación

La usabilidad puede definirse como “la capacidad que tiene un producto software para ser entendido, aprendido, operable, atractivo para el usuario y confirme a estándares, cuando es utilizado bajo una condiciones específicas” (ISO/IEC 9126-1: 2001).

Un método eficaz para medir la usabilidad de un producto consiste en realizar un test de usuarios mediante la observación de cómo un grupo de personas lleva a cabo una serie de tareas determinadas. El evaluador debe registrar todos y cada uno de los problemas con los que el usuario se vaya encontrando, anotando sus reacciones y comentarios para poder analizarlos posteriormente.

Una prueba de usabilidad no debe ser pospuesta hasta las etapas finales del desarrollo del proyecto: debe ir realizándose sobre los prototipos que se vayan generando para descubrir el mayor número de errores posibles en el diseño y poder así resolverlos en futuras versiones del proyecto. Además, los expertos en la materia indican que para cada una de estas pruebas no es necesaria una gran cantidad de usuarios, admitiendo alrededor de 5 usuarios por test.

En un test de usabilidad con usuarios no sólo será importante analizar el transcurso del mismo, sino tener en cuenta una primera impresión de éste ante el diseño dado. Por este motivo, se recomienda realizar un primer acercamiento al objeto de estudio para poder juzgar la reacción del usuario y seguidamente comenzar con la ejecución del conjunto de tareas.

Las pruebas diseñadas deben ajustarse a cualquier tarea típica que un usuario real llevaría a cabo, evitando que sean demasiado genéricas y asegurando que sean factibles y de una duración razonable. Durante el proceso el evaluador debe analizar todos los errores que el usuario cometa, anotar el tiempo empleado en cada tarea así como el grado de satisfacción final. Si es posible se debe solicitar al participante que exprese sus pensamientos en voz alta para detectar la razón de sus acciones y las cuestiones que no entienda. Una vez finalizados

los tests, los datos recogidos deben ser sintetizados en un informe de usabilidad con el fin de mejorar el diseño en base a los resultados obtenidos.

6.2 Pruebas realizadas

Antes de comenzar las tareas que se deben resolver, realiza las siguientes acciones y responde:

- Ejecuta la aplicación: VISTac-IMAS-Can,
- Navega por la aplicación
- Observa y explica para qué crees que sirve la aplicación

Tareas a resolver:

1. Consulta la información sobre la “Energía eléctrica disponible”.
2. Analiza la evolución de este indicador en Tenerife y Gran Canaria, durante los últimos 5 años.
3. Realiza una comparativa de población incluyendo la población turística entre las islas menores durante un periodo que incluya todos los datos anuales disponibles.
4. Analiza los datos del índice de motorización desde el año 2010.
5. Averigua el porcentaje de superficie que aporta Fuerteventura al total de Canarias.
6. ¿Qué isla es la que tiene una mayor tasa de ocupación turística?
7. ¿De qué periodo se tiene información de la concentración de ganado por superficie? ¿Qué isla es la que cuenta con más ganado porcino?

6.3 Resultados obtenidos

Primera impresión

Una vez que los usuarios han ejecutado la aplicación y ha identificado la lista de categorías de la pantalla inicial, sólo experimentando con ellas detecta que la aplicación permite consultar datos estadísticos sobre indicadores que están agrupados en las categorías que se observan. No existe ninguna indicación, icono, opción de ayuda, que muestre el propósito de la misma, por lo que inicialmente se sienten un poco desconcertados.

Tareas

- A lo largo del desarrollo de las tareas, se detectan ciertos aspectos de diseño que podrían mejorarse, para facilitar el manejo de la aplicación el análisis de los datos y corregir errores detectados. Los problemas encontrados por los usuarios quedan resumidos en los siguientes puntos: No existen ayudas: Breve información inicial del propósito de la aplicación, resumen del manejo de la misma, breve información respecto al indicador que se seleccione.
- La pantalla de selección de las categorías temporal y territorial es desconcertante, se desconoce si obligatoriamente tienen que seleccionarse alguna isla o periodo de tiempo.
- Es necesario hacer *scroll* de varias pantallas para poder continuar, una vez que se han elegido (o no), las islas y años sobre los que se consulta.
- No existe una forma rápida de situarse en el inicio de la pantalla ni de la aplicación.
- Los indicadores en los que se utiliza la superficie para los cálculos no se genera la serie de datos, solo el dato correspondiente al año 2007.
- Una vez que se realiza una consulta, no se muestra en ningún momento qué datos se están representando.
- Se produce confusión con respecto a las unidades, en algunos casos no se muestran y en otros parecen erróneas.
- En ocasiones se debe pulsar sobre el nombre del indicador y en otras sobre el botón “consulta” para que se lleve a cabo la descarga de los datos.

- Cuando los datos representan una contribución al total de Canarias, es más adecuado un gráfico de sectores.
- En gráficos de indicadores que representan valores absolutos sería las islas menores, sobre todo El Hierro y La Gomera se convierten prácticamente en una línea recta por las diferencias de escala con las capitalinas.

Todos los cambios que los usuarios proponen quedan reflejados más explícitamente en el apartado 6.4. Propuesta de cambios.

6.4 Propuesta de cambios

6.4.1 Cambios en la disposición de elementos

Una vez se llega a la pantalla en la que se permite filtrar la consulta, resultaría más cómodo que los parámetros geográficos y temporales estuvieran seleccionados todos por defecto, y en caso de que el usuario quiera excluir algún parámetro de la consulta, realice la acción manualmente. Otro elemento que debe cambiar su disposición es el botón que permite ejecutar la consulta, ya que debería estar a la cabeza del formulario de filtrado, y no al final del mismo. Con esto, se evita que el usuario tenga que desplazarse por todo el listado, que podría llegar a tener un número considerable de opciones, para ejecutar la consulta.

Una vez se muestran los resultados, la vista por defecto debería ser la de la visualización gráfica, ya que ofrece una idea más general de los resultados, y en caso de querer realizar una consulta rápida, esta vista permite captar más rápidamente variaciones entre valores, tendencias, etc.

Por último, en la visualización numérica de los datos, resulta algo complicado de observar los valores con la disposición actual de los datos. El diseño de esta vista mejoraría si los datos se ofreciesen en forma de tabla, en la cual sus filas fueran los años consultados, y cuyas columnas fuera el dato correspondiente y su relación con el total de Canarias, ya sea en forma de porcentaje o mostrando el dato total referente a la comunidad autónoma. Otra posibilidad es utilizar una interfaz en la que puedas desplegar los datos de una isla o los datos de un año según prefiera el usuario.

Incluir una opción de volver al inicio y otra de subir al comienzo de la pantalla.

Sustituir el encabezado muestre el texto: “Resultado de la consulta” por el nombre del indicador que se haya seleccionado.

Incluir una pantalla inicial de ayuda con el propósito de la aplicación y una explicación breve del manejo de la misma. En el momento en que se selecciona un indicador mostrar también una breve información del mismo.

6.4.2 Cambios para la reducción de elementos intrusivos

Algunos elementos de la aplicación podrían tener la opción de contraerse y expandirse, con la finalidad de que si ocupa una gran proporción de la pantalla no resulten intrusivos.

Algunos de estos elementos son las opciones de las islas y los años en la vista del formulario de filtrado de la consulta o el conjunto de datos numéricos para cada territorio insular.

6.4.3 Mejoras en el formato numérico

Actualmente, los datos numéricos vienen mostrados con la parte entera sin separación, y en caso de ser un valor con parte decimal, se separa la parte entera de la fraccionaria con un punto decimal.

Podría hacerse una separación en la parte entera de los valores en los millares, de forma que los valores elevados puedan ser más fácilmente legibles.

6.4.4 Mejoras en la selección de opciones del filtro de consulta

Aunque actualmente basta con no seleccionar ninguna opción para interpretar que la consulta debe incluirlas todas, resultaría muy interesante y más intuitiva la adición de un botón que permite seleccionar todas o ninguna de las opciones en el formulario de filtro de consulta, tanto para las islas como para los años, en cuyo caso, el listado puede tener bastante opciones. Otra opción es seleccionar sólo un año y que se muestren los datos desde ese año hasta el actual.

Permitir la opción de distinguir entre: “Islas Capitalinas”, “Lanzarote, Fuerteventura y La Palma”, “La Gomera y El Hierro”, en aquellos indicadores en que se aconseje por las diferencias de escala.

Proporcionar la opción de gráfico de sectores en los casos en que sea recomendable.

Parte III.

Conclusiones

Capítulo 7.

Conclusiones y líneas futuras

Durante el desarrollo de este proyecto, he podido tomar conciencia de la importancia de la visualización de datos y de su correcta representación, dado que la precisión resulta un factor fundamental y puede marcar la diferencia, sobre todo si esos datos van a ser la base de una toma de decisiones.

Se ha prestado especial atención a la granularidad de los datos que se ofrecen, para que la representación sea lo más cómoda posible para el público objetivo de la aplicación.

En general, desarrollar esta aplicación ha sido una experiencia muy enriquecedora, no sólo por la aplicación del conocimiento ya adquirido de HTML5, CSS3 y JavaScript; también por la ampliación del mismo al utilizar *frameworks* como AngularJS que requiere cierta formación complementaria. También ha resultado interesante el análisis de la usabilidad de la aplicación y cómo el propio usuario de la aplicación puede aportar sus ideas tras probar el software, y así contribuir al desarrollo y descubrir nuevas perspectivas de uso de la aplicación que el desarrollador pudo no haber considerado en primera instancia.

Otra característica que favorece a la aplicación, tanto en el desarrollo como en su futuro mantenimiento es el uso de la API. Esta herramienta fundamental permite añadir o modificar los datos que maneja la aplicación únicamente modificándolas en el servidor de la API, y eso permite que los nuevos datos puedan ser visualizados en la aplicación sin necesidad de que el desarrollador tenga que lanzar una actualización o algún otro tipo de parche. Observando la expansión de este tipo de sistemas, puedo afirmar que se ha trabajado sobre un sistema innovador y con vistas de futuro.

Por último, en cuanto a posibles líneas futuras de trabajo, podría considerarse la adición de características como la comparación simultánea de varios indicadores, siempre y cuando estos sean compatibles; la conexión de la

aplicación con redes sociales o la posibilidad de compartir las consultas, lo que expandiría enormemente la distribución de los datos entre grupos de trabajo, por ejemplo; o también, la adaptación de esta aplicación llevada a otras áreas en las que pueda ser interesante observar datos estadísticos.

Capítulo 8.

Summary and Conclusions

During this project's development, I have been conscious about the importance of data visualization and its correct representation, insomuch as precision can become an essential factor and can makes the difference if such data are going to be the base of a decision making.

Offered data granularity have been watched carefully, allowing to make the representation as comfortable as possible to the application's target public.

In the main, developing this application has been a very valuable experience, not only for being able to apply the previous knowledge about HTML5, CSS3 or JavaScript; also for learning to use frameworks like AngularJS which requires some further training. Usability analysis has been also interesting because the user can contribute with ideas after testing the software, and being part of the development, as much as the user can offer new perspectives that developers could have not realized of on a first view.

Another useful feature, both development and maintenance, of this application is the implication of using an API. This fundamental tool allows to add or modify data just on the server, and those changes are automatically applied, no needing that developers launch any update or patch. Looking at the spreading of these type of systems, is a fact that this project has been developed onto an innovative technology.

By last, some many future improvements can be the addition of features like the simultaneous comparison of two or more compatible indicators; connecting the data visualization with social networks, allowing the data distribution through workgroup's members, for example; or the rearranging of the application to offer statistics data to different areas of interest.

Capítulo 9.

Presupuesto

Basados en el cronograma de trabajo de este proyecto, y en una estimación de los costes para su desarrollo ha elaborado el siguiente presupuesto:

- Se precisará del servicio de un analista, que realizará el trabajo de investigación que supone obtener la lista de indicadores referidos al área de medioambiente, y la mejor forma de disponerlos en la aplicación, según las preferencias del público objetivos de la aplicación.
 - Coste estimado: 15,92€/hora laboral.
- Se precisará de un desarrollador software, que se encargará de la realización de la aplicación móvil, de generar los diferentes prototipos que se usarán en los tests de usabilidad, y de mejorar el producto según los resultados de estas pruebas.
 - Coste estimado: 12,63€/hora laboral.
- Se precisará de un usuario capaz de probar la aplicación y superar el test de usabilidad.
 - Coste estimado: 10,00€/sesión.

Según las líneas del cronograma, cada tarea asignada a cada uno de los seleccionados anteriormente consta de las siguientes horas laborales o sesiones:

Estimación temporal del desarrollo del proyecto	
Tipo de actividad	Tiempo o sesiones dedicados
Análisis: Revisión de antecedentes y estado del arte, consulta de información a la API ISTAC, visualización de la información.	8 semanas
Desarrollo software: Desarrollo de los prototipos para Android e iOS, correcciones tras las pruebas de usabilidad.	6 semanas
Pruebas de usabilidad: Ejecución del test de usabilidad	1 sesión

Tabla 9.1 Estimación temporal del desarrollo del proyecto

Se asume que cada jornada laboral es de ocho horas, y que durante cada semana hay cinco jornadas laborales. Asumiendo los costes de los servicios de los empleados, más los gastos que deben realizarse para poner la aplicación disponible en las tiendas de aplicaciones para Android e iOS, el presupuesto de este proyecto es el siguiente:

Presupuesto total del proyecto	
Concepto	Precio
Salario analista (15,92€ x 8 horas x 40 jornadas laborales)	5.094.40 €
Salario desarrollador software (12,63€ x 8 horas x 30 jornadas laborales)	3.031.20 €
Incentivo usuario pruebas usabilidad (10,00€ x 1 sesión)	10.00 €
Licencia de desarrollador de Google – Permite subir aplicaciones para Android a la Play Store	22.72 €
Licencia de desarrollador de Apple – Permite subir aplicaciones para iOS a la App Store	89.95 €
TOTAL	8.248,27 €

Tabla 9.2 Presupuesto total del proyecto

Apéndice A.

Creación de gráficas con D3

A.1. Ejemplo: Gráfico de líneas

```
/******  
*  
* linechart_example.js  
*  
*****  
*  
* AUTOR: Eliasib Jesús García Martín  
*  
*  
* FECHA: 07/07/2015  
*  
*  
* DESCRIPCION: Creación de un gráfico de líneas usando la librería D3.js (se presupone  
*             previamente importada) sobre un set de datos de ejemplo.  
*  
*  
*****/  
  
// Set de datos de ejemplo. En el caso real del proyecto, se usa una estructura  
// idéntica con los datos reales obtenidos tras una consulta.  
var data_set = [{  
    "dato": 140.35,  
    "fecha": 2010  
},  
{  
    "dato": 138.65,  
    "fecha": 2011  
},  
{  
    "dato": 135.4,  
    "fecha": 2012  
}];
```

```

var grafica = d3.select('#grafica'), // Elemento donde se plasmará la gráfica
    WIDTH = grafica.width(), // Ancho de la gráfica (en píxeles)
    HEIGHT = grafica.height(), // Alto de la gráfica (en píxeles)
    MARGINS = {
        top: 20, // Píxeles de margen superior
        right: 20, // Píxeles de margen a la derecha
        bottom: 20, // Píxeles de margen inferior
        left: 50 // Píxeles de margen a la izquierda
    },

/*
 * Variable que refleja la escala del eje x, basada en el largo del eje,
 * que irá desde el margen izquierdo hasta la longitud de la gráfica, menos
 * el margen derecho; y en el dominio del rango de datos, que será desde la
 * primera fecha en el array hasta la última, dado que los datos se organizan
 * ordenadamente según la fecha.
 */

xScale = d3.scale.linear().range([MARGINS.left, WIDTH - MARGINS.right]).domain(
    [parseInt(data_set[0].fecha),
     parseInt(data_set[data_set.length-1].fecha)]),

/*
 * Variables auxiliares para la búsqueda del dato mayor y menor.
 */

maxDato = -Infinity, minDato = Infinity;

// Búsqueda de los datos de mayor y menor valor
for (var i = 0; i < data_set.length; i++) {
    var dato_float = parseFloat(data_set[i].dato);
    if (dato_float > maxDato)
        maxDato = dato_float;
    if (dato_float < minDato)
        minDato = dato_float;
};

/*
 * Variable que refleja la escala del eje y, basada en el alto del eje,
 * que irá desde el alto de la gráfica menos el margen superior, hasta el margen
 * inferior; y en el dominio del rango de datos, que será desde la
 * el dato de menor valor hasta el dato de mayor valor.
 */

var yScale = d3.scale.linear().range([HEIGHT - MARGINS.top, MARGINS.bottom])
    .domain([minDato,maxDato]);

```

```

/* Creación de los ejes horizontal y vertical. Los "ticks" hacen referencia al número
 * deseado de líneas separadoras en los ejes. D3 generará estos valores separadores en
 * función al rango de datos y el número de muestras.
 * En el caso del eje y, debe orientarse hacia la izquierda, para poder verse dispuesto
 * en vertical.
 */
var xAxis = d3.svg.axis().scale(xScale).tickFormat(d3.format("d")).ticks(6),
    yAxis = d3.svg.axis().scale(yScale).orient("left").tickFormat(d3.format("d"))
        .ticks(6);

// Se añaden los ejes a la gráfica
grafica.append("svg:g").attr("class","axis").attr("transform", "translate(0, " +
    (HEIGHT - MARGINS.bottom) + ")").call(xAxis);
grafica.append("svg:g").attr("class","axis").attr("transform", "translate(" +
    (MARGINS.left) + ",0)").call(yAxis);

// Declaración de la función generadora de la línea de la gráfica
var lineGen = d3.svg.line()
    .x(function (d) {
        return xScale(d.fecha); // Posición a lo largo del eje x que tomará
        // el punto d, en función de su fecha
    })
    .y(function (d) {
        return yScale(d.dato); // Posición a lo largo del eje y que tomará
        // el punto d, en función de su dato
    });

// Se genera la línea, recorriendo el set de datos para definir sus puntos, y se añade
// a la gráfica
for (var i = 0; i < data_set.length; i++) {
    grafica.append('svg:path')
        .attr('d', lineGen(data_set[i])) // Referencia al punto a representar
        .attr('stroke', 'blue') // Color del trazo, azul en este
        // ejemplo */
        .attr('stroke.width', 2) // Ancho del trazo, en píxeles
        .attr('fill', none); // Color de relleno (útil si fuera
        // una figura con superficie, pero
        // irrelevante en una línea) */
};

/***** END OF FILE *****/

```

Bibliografía

- [1] Sistemas de Indicadores Medio Ambientales de Canarias (SIMAC) <http://www.gobiernodecanarias.org/medioambiente/piac/temas/participacion-informacion-publica/simac/sistema-indicadores-ambientales/>
- [2] Apache Cordova (Framework de desarrollo de aplicaciones) <https://cordova.apache.org/>
- [3] Phonegap (Framework de desarrollo de aplicaciones) <http://phonegap.com/>
- [4] Appcelerator (Framework de desarrollo de aplicaciones) <http://www.appcelerator.com/>
- [5] React Native (Framework de desarrollo de aplicaciones) <https://facebook.github.io/react-native/>
- [6] Ionic Framework (Framework de desarrollo de aplicaciones) <http://ionicframework.com/>
- [7] AngularJS (Framework MVC para JavaScript) <https://angularjs.org/>
- [8] jQuery (Librería JavaScript de manipulación DOM) <https://jquery.com/>
- [9] Zepto.js (Librería Javascript de manipulación DOM) <http://zeptojs.com/>
- [10] Documentación API ISTAC <http://www.gobiernodecanarias.org/istac/indicators/api/indicators/v1.0#!/indicators>
- [11] JavaScript Object Notation (JSON) (Formato de datos) <http://json.org/>
- [12] Swagger (Framework para representación con interfaz de APIs) <http://swagger.io/>
- [13] Despliegue del Trabajo de Fin de Grado del Grado en Ingeniería Informática: Título: Pulso Económico (PULSEC). Visualización de información en Firefox OS, Autor: Beneharo González González, Director: Isabel Sánchez Berriel, Año: 2014. <http://banot.etsii.ull.es/alu4240/pulsec/index.html>

- [14] Web Consulta Indicadores Económicos EUROSTAT
<http://ec.europa.eu/eurostat/cache/infographs/economy/desktop/index.html>
- [15] Web Creador Widgets Indicadores Económicos del ISTAC
<http://www.gobiernodecanarias.org/istac/indicators/widgets/creator?type=temporal>
- [16] Web Proyecto Interreg III-B INDICE
http://www.gobiernodecanarias.org/cmayot/interreg/indice/observatorio_macaronesia/indice/index.html
- [17] Referencia sobre XML <http://www.w3schools.com/xml/>
- [18] Referencia sobre CSV https://en.wikipedia.org/wiki/Comma-separated_values
- [19] LibreOffice Calc (Hojas de cálculo con herramienta de creación de gráficas)
<https://es.libreoffice.org/descubre/calc/>
- [20] Hojas de Cálculo de Google (Hojas de cálculo con herramienta de creación de gráficas) https://www.google.com/intl/es_es/sheets/about/
- [21] Microsoft Office Excel (Hojas de cálculo con herramienta de creación de gráficas) <https://products.office.com/es-mx/excel>
- [22] Tableau Public (Software de visualización de datos)
<https://public.tableau.com/s/>
- [23] Data-Driven Documents (D3) (Librería Javascript de visualización de datos) <http://d3js.org/>
- [24] Google Developers Chart Tools (Herramienta de gráficos para desarrolladores de Google) <https://developers.google.com/chart/?hl=es>
- [25] Referencia a la tecnología AJAX <http://www.w3schools.com/ajax/>
- [26] WebStorm (IDE para proyectos web)
<https://www.jetbrains.com/webstorm/>
- [27] Documentación del emulador de Android (Android Developers)
<http://developer.android.com/tools/help/emulator.html>

- [28] Documentación del simulador de iOS (iOS Developer Library) https://developer.apple.com/library/ios/documentation/IDEs/Conceptual/iOS_Simulator_Guide/
- [29] Git (Herramienta de control de versiones) <https://git-scm.com/>
- [30] GitHub (Repositorios remotos públicos) <https://github.com/about>
- [31] Trello (Herramienta de productividad) <https://trello.com/home>
- [32] Cross-Domain Solution (Artículo en Wikipedia) https://en.wikipedia.org/wiki/Cross-domain_solution
- [33] JSON With Padding (JSONp) <http://json-p.org/>
- [34] Extensión “Allow-Control-Allow-Origin: *” para Chrome <https://chrome.google.com/webstore/detail/allow-control-allow-origi/nlfbmbojpeacfhkpbjhddihlkkiljbi?hl=es>

