

**Universidad Internacional de La Rioja (UNIR)**

**Escuela de Ingeniería**

**Grado en Ingeniería Informática**

# Aplicación web para el análisis estadístico de datos de la calidad del agua y del aire

**Ubicación del código fuente:**

<https://github.com/merclsb/AEPRO>

**Trabajo Fin de Grado**

**Presentado por:** Sánchez Romero, Alberto

**Director:** Pedraza Gómara, Luis

Ciudad: Zaragoza

Fecha: 14/09/2017

## Agradecimientos

En primer lugar, quiero agradecerle a Mariola la paciencia que ha tenido a lo largo de estos meses, y que me ha permitido concluir este proyecto. En segundo lugar, a mis padres por el apoyo que siempre me han transmitido. De igual manera, a Javier Martínez y a Joaquín Sancho, por motivarme y permitir que les ayude en este proyecto. Y por último a mí director Luis Pedraza Gómara por dedicarme parte de su tiempo.

*Gracias a todos.*

## Resumen

El objetivo de este trabajo fin de grado es el desarrollo de una aplicación web que permita a los usuarios extraer parámetros significativos, para la identificación de anomalías, en los datos obtenidos de procesos de medida de variables medioambientales. Para ello se validará la información, se analizará y se mostrará mediante gráficos que permitan identificar de una manera rápida y precisa si existe alguna alteración. El desarrollo de esta aplicación web se enmarca como tarea específica de un proyecto de investigación de control de parámetros medioambientales desarrollado en el Centro Universitario de la Defensa. Para la consecución de este desarrollo, se usará Python como lenguaje principal haciendo uso del framework de desarrollo Django para la construcción backend, junto con Bootstrap en el frontend, así como un módulo de análisis estadístico, implementado en Matlab por investigadores del proyecto, y del software de visualización de datos Plotly.

**Palabras Clave:** Análisis Estadístico, Python, Django, Plotly, Calidad del aire, Calidad del agua

## Abstract

The objective of this work end of degree is the development of a web application that allows users to extract significant parameters, for the identification of anomalies, in the data obtained from environmental variables measurement processes. For this purpose, the information will be validated, analyzed and displayed by means of graphs that allow rapid and accurate identification of any alterations. The development of this web application is framed as a specific task of a research project to control environmental parameters developed at the Centro Universitario de la Defensa.

To achieve this development, Python will be used as the main language using the Django development framework for backend construction, together with Bootstrap in the frontend, as well as a statistical analysis module, implemented in Matlab by researchers of the project, and the Plotly data visualization software.

**Keywords:** Statistical Analysis, Python, Django, Plotly, Air quality, Water quality



# Índice

<b>1. Introducción .....</b>	<b>1</b>
<b>1.1. Motivación.....</b>	<b>1</b>
<b>1.2. Objetivos .....</b>	<b>2</b>
<b>1.3. Estructura del documento.....</b>	<b>2</b>
<b>2. Contexto y estudio preliminar .....</b>	<b>4</b>
<b>2.1. Dominio de aplicación.....</b>	<b>4</b>
<b>2.2. Herramientas para el análisis estadístico de datos.....</b>	<b>5</b>
2.2.1. Sistemas de muestreo.....	5
2.2.2. R-Markdown .....	7
2.2.3. Matlab Web Browser .....	8
<b>2.3. Herramientas para el desarrollo de aplicaciones web .....</b>	<b>9</b>
2.3.1. Lenguajes de programación .....	9
2.3.2. Framework de desarrollo.....	10
2.3.2.1. Patrón Modelo Vista Controlador.....	10
2.3.2.2. Frameworks de desarrollo para aplicaciones web.....	12
2.3.3. Base de datos .....	15
2.3.3.1. Sistema gestor de base de datos .....	15
2.3.3.2. Object-Relational Mapper .....	15
2.3.4. Framework CSS.....	17
2.3.5. Visualización de datos.....	20
2.3.5.1. Estándares web para visualización de datos.....	20
2.3.5.2. Librerías para gráficos 2D.....	23
<b>3. Identificación de requisitos .....</b>	<b>29</b>
<b>3.1. Especificación de requisitos software .....</b>	<b>29</b>
3.1.1. Requisitos funcionales .....	29
3.1.2. Requisitos No funcionales .....	39
<b>3.2. Gestión del proyecto .....</b>	<b>41</b>
3.2.1. Ciclo de vida.....	41
3.2.2. Modelo de desarrollo en cascada.....	41
3.2.3. Modelo iterativo-incremental .....	43
<b>3.3. Herramientas empleadas para el desarrollo del proyecto .....</b>	<b>44</b>
3.3.1. Entornos virtuales.....	44

3.3.2. Pycharm.....	44
3.3.3. Github .....	45
<b>3.4. Planificación .....</b>	<b>46</b>
<b>4. Diseño de la solución.....</b>	<b>48</b>
4.1. Diseño global del sistema.....	48
4.2. Modelo de datos .....	49
4.3. Diagrama de actividad.....	51
4.4. Mapa de navegación.....	56
4.5. Diseño de la interfaz de usuario .....	57
<b>5. Descripción técnica de la solución .....</b>	<b>61</b>
5.1. Descripción funcional de la aplicación .....	61
5.2. Tecnologías utilizadas en la implementación.....	65
5.3. Arquitectura de la aplicación .....	66
5.3.1.1. Implementación del patrón Modelo Vista Template.....	66
5.3.1.2. Arquitectura Django .....	67
5.4. Modelo de datos .....	72
5.5. Implementación de la interfaz de usuario .....	73
<b>6. Validación técnica de la solución.....</b>	<b>76</b>
6.1. Pruebas de cada subsistema.....	76
6.2. Pruebas de aceptación.....	76
<b>7. Conclusiones y líneas de trabajo futuro.....</b>	<b>81</b>
7.1. Conclusiones .....	81
7.2. Líneas de trabajo futuras .....	83
<b>Bibliografía.....</b>	<b>85</b>
<b>Anexo I. Manual de usuario .....</b>	<b>88</b>
<b>Anexo II. Prototipos de la interfaz de usuario .....</b>	<b>104</b>
<b>Anexo III. Pruebas de subsistema.....</b>	<b>110</b>

## Índice de figuras

Figura 1. Registros obtenidos por una estación de medición. Red SAICA.....	6
Figura 2. Registros obtenidos por una estación de medición. Ayto ZGZ.....	6
Figura 3. Ejemplo de contenido generado R-Markdown .....	7
Figura 4. Contenido web generado con Matlab que muestra la curva de la función seno.....	8
Figura 5. Patrón de interacción de componentes en una arquitectura MVC. Fuente [13]. ..	11
Figura 6. Diagrama de secuencia del patrón MVC donde se puede observar el flujo de control durante una solicitud http. ....	12
Figura 7. Ranking basado en la popularidad de los desarrolladores. Imagen obtenida de la página Hotframework . ....	13
Figura 8. ORM provee de un “puente” entre las tablas, relaciones, campos de las bases de datos relacionales y los objetos Python.....	16
Figura 9. La imagen muestra la funcionalidad responsive web design, que permite poder adaptar el contenido de una página al dispositivo que lo está visualización. Imagen obtenida de la página Materializecss .....	18
Figura 10. Ejemplo de la variedad de la paleta de gráficos que pueden implementarse con D3. Imagen obtenida de: <a href="https://github.com/d3/d3/wiki/Gallery">https://github.com/d3/d3/wiki/Gallery</a> . ....	24
Figura 11. Ejemplo de los gráficos a usar con Google Charts. Imagen obtenida de: <a href="https://developers.google.com/chart/interactive/docs/gallery">https://developers.google.com/chart/interactive/docs/gallery</a> .....	25
Figura 12. Ejemplo de grafico creado con Chart.js. Fuente . ....	26
Figura 13. Ejemplo de grafico creado con Highcharts. Fuente .....	27
Figura 14. Clasificación de la paleta de gráficos soportados por Plotly.....	28
Figura 15. Diagrama de caso de uso: Sistema de Análisis. ....	35
Figura 16. Diagrama de Caso de uso: Sistema de registro y autenticación de usuarios.....	39
Figura 17. Fases que componen el modelo de desarrollo en cascada. ....	42
Figura 18. Ciclo de vida iterativo e incremental, en donde cada fase (iteración) sirve de base para la siguiente.....	43
Figura 19. Ejemplo de stack usado en Virtualenv.....	44
Figura 20. Ejemplo de cómo Pycharm es capaz de completar los comandos.....	45
Figura 21. Ejemplo de la aplicación para escritorio Github .....	46
Figura 22. Diagrama de Gantt para la planificación de proyecto .....	47
Figura 23. Aproximación del diseño global del sistema .....	49
Figura 24. Diagrama Entidad-Relación AEPRO .....	50
Figura 25. Diagrama de actividad: Proceso de creación de un nuevo análisis.....	52

Figura 26. Diagrama de actividad: Proceso de validación de un fichero.....	53
Figura 27. Diagrama de actividad: Proceso de visualización de análisis realizados y los resultados obtenidos .....	54
Figura 28. Diagrama de actividad: Proceso de alta y gestión de usuarios.....	55
Figura 29. Mapa navegación del sitio web,.....	56
Figura 30. Estructura de la interfaz de usuario.....	57
Figura 31. Mockup de la página para crear nuevo análisis .....	58
Figura 32. Mockup de la página que muestra listado de análisis creados.....	59
Figura 33. Mockup Resultados del análisis funcional de datos en un dispositivo móvil .....	60
Figura 34. Ejemplo de un Boxplot(izq) y de un histograma (drch).....	62
Figura 35. Ejemplo de un gráfico XBAR: superior representación de las medias. Inferior representación de las desviaciones.....	62
Figura 36. Resultado de un análisis funcional. En negro las curvas que representan los días en los que se ha producido una desviación en las emisiones de gas. [4] .....	63
Figura 37. Diagrama general de la metodología, en donde se puede ver las diferentes fases del proceso de análisis. [34].....	64
Figura 38. Diagrama general del sistema con las tecnologías aplicadas .....	65
Figura 39. Funcionamiento del MTV de Django.....	67
Figura 40. Diagrama del flujo de una solicitud en Django.....	68
Figura 41. Diagrama de interacción empleado para el proyecto principal (WEBSITE).....	69
Figura 42. Diagrama de interacción empleado para la aplicación (AEPRO) .....	70
Figura 43. Diagrama del modelo de datos de la aplicación.....	72
Figura 44. Vista del apartado: crear nuevo análisis.....	73
Figura 45. Vista del apartado: lista de resultado.....	74
Figura 46. Vista del apartado: resultado de análisis funcional de datos.....	74
Figura 47. Vista del apartado. Diseño <i>responsive</i> o adaptativo de la aplicación.....	75
Figura 48. Landing <i>page</i> de la aplicación AEPRO.....	88
Figura 49. Página de registro para un nuevo usuario.....	89
Figura 50. Mensaje de registro correcto para el usuario.....	89
Figura 51. Email enviado al usuario para la activación de la cuenta.....	90
Figura 52. Página de inicio de la aplicación AEPRO.....	91
Figura 53. Creación de nuevo análisis con la información aportada por el usuario.....	93
Figura 54. Mensaje de creación correcta de un nuevo análisis.....	93
Figura 55. Lista de análisis creados y el estado de procesamiento.....	94
Figura 56. Lista de resultados: análisis pendiente de procesar.....	94

Figura 57. Lista de resultados: análisis calculando.....	95
Figura 58. Lista de resultados: análisis resultado disponible.....	95
Figura 59. Lista de resultados: error durante el proceso de análisis.....	95
Figura 60. Mensaje previo al borrado de la información correspondiente a un error durante el proceso de análisis.....	96
Figura 61. Resultado de un análisis FDA.....	97
Figura 62. Resultado de un análisis FDA: manipulación de los gráficos.....	97
Figura 63. Apartado para actualizar la información de los resultados FDA.....	98
Figura 64. Mensaje de confirmación de borrado de análisis.....	98
Figura 65. Resultado de un análisis CEP.....	99
Figura 66. Sección para validar el formato del fichero.....	100
Figura 67. Respuesta para un formato correcto.....	100
Figura 68. Respuesta para un formato incorrecto.....	100
Figura 69. Menú de acceso al perfil de usuario.....	101
Figura 70. Información del perfil de usuario.....	101
Figura 71. Menú para el cambio de contraseña.....	102
Figura 72. Formulario de contacto.....	102
Figura 73. Error de login en caso de no recordar la contraseña.....	103
Figura 74. Email proporcionado por el usuario para restablecer contraseña.....	103
Figura 75. Mensaje de envío de instrucciones para restablecimiento de contraseña.....	103
Figura 76. Mockup: Landig page.....	104
Figura 77. Mockup: pagina de registro en la aplicacion.....	105
Figura 78. Mockup: página de <i>login</i> en la aplicación.....	106
Figura 79. Mockup Detalle de un análisis.....	107
Figura 80. Mockup resultado análisis Control Estadístico de procesos.....	108
Figura 81. Mockup resultado análisis: Análisis Funcional de Datos.....	109
Figura 82. Página de registro.....	110
Figura 83. Email recibido para la activación de la cuenta.....	111
Figura 84. Mensaje de activación de la cuenta.....	111
Figura 85. <i>Landing page</i> de acceso a la aplicación.....	112
Figura 86. Página de <i>login</i> en caso de producirse algún error.....	112
Figura 87. Página de inicio para usuarios autenticados.....	113
Figura 88. Menú desplegable para el perfil de usuarios.....	113
Figura 89. Información del usuario.....	114
Figura 90. Formulario para la creación de un nuevo análisis.....	115

Figura 91. Mensaje de análisis creado correctamente.....	115
Figura 92. Lista de resultados. ....	116
Figura 93. Gráficos resultantes para un análisis FDA.....	116
Figura 94. Gráficos resultantes para un análisis CEP.....	117
Figura 95. Apartado de validación de formato de ficheros.....	118
Figura 96. Mensaje resultante de la validación del formato de un fichero.....	118

## Índice de tablas

Tabla 1. Lenguajes de programación más usados en el mundo . + indica una subida de puesto y – indica una bajada de puesto. ....	9
Tabla 2. Tabla comparativa de características entre Django y Flask. ....	15
Tabla 3. Comparativa de características: SQLite, Mysql y PostgreSQL.....	15
Tabla 4. Comparativa de las características más destacables de Bootstrap y <i>Foundation</i> ..	19
Tabla 5. Tabla comparativa de las características más destacables de Canvas frente a SVG. <b>Error! Bookmark not defined.</b> .....	22
Tabla 6. Librerías para gráficos 2D .....	24
Tabla 7. RQF 1 – Crear nuevo análisis.....	30
Tabla 8. RQF 2 – Selección del tipo de análisis. ....	30
Tabla 9. RQF 3 – Selección del periodo temporal. ....	30
Tabla 10. RQF 4 – Fichero con información a analizar.....	31
Tabla 11. RQF 5 – Fichero con información a analizar.....	31
Tabla 12. RQF 6 – Validar fichero. ....	31
Tabla 13. RQF 7 – Borrar análisis .....	32
Tabla 14. RQF 8 – Listar análisis creados.....	32
Tabla 15. RQF 9 – Pagar la lista de resultados. ....	32
Tabla 16. RQF 10 – Filtrar la lista de resultados.....	33
Tabla 17. RQF 11 – Indicar el estado del proceso de análisis.....	33
Tabla 18. RQF 12 – Acceder a información análisis.....	33
Tabla 19. RQF 13 – Actualizar la información de un análisis.....	33
Tabla 20. RQF 14 – Resultado de análisis CEP.....	34
Tabla 21. RQF 15 – Resultado de análisis FDA.....	34
Tabla 22. RQF 16 – Descargar los resultados generados.....	34
Tabla 23. RQF 17 – Registro de un nuevo usuario.....	36
Tabla 24. RQF 18 – Activación de la cuenta de usuario.....	36
Tabla 25. RQF 19 – Sistema de autenticación.....	36
Tabla 26. RQF 20 – Restaurar contraseña.....	37
Tabla 27. RQF 21 – Perfil y contenido de usuario.....	37
Tabla 28. RQF 22 – Formulario de contacto.....	37
Tabla 29. RQF 23 – Panel Staff.....	38
Tabla 30. RQNF 1 – Comunicación con MATLAB.....	39
Tabla 31. RQNF 2 – Información asociada a cada usuario.....	40

Tabla 32. RQNF 3 – Diseño de interfaz adaptativa.....	40
Tabla 33. RQNF 4 – Validaciones de los campos de los formularios.....	40
Tabla 34. RQNF 5 – Validaciones sobre ficheros. ....	40
Tabla 35. RQNF 6 – Datetimepicker.....	41
Tabla 36. Entidad Usuario y sus atributos. ....	50
Tabla 37. Entidad Análisis y sus atributos .....	51
Tabla 38. Entidad ResultadoFDA y sus atributos .....	51
Tabla 39. Entidad ResultadoCEP y sus atributos .....	51
Tabla 40. Descripción y requisitos asociados a la interfaz de creación de un nuevo análisis. .....	58
Tabla 41. Descripción y requisitos asociados al listado de análisis generados.....	59
Tabla 42. Descripción y requisitos asociados a la interfaz de resultados de un análisis FDA visto en un dispositivo móvil. ....	60
Tabla 43. Analogía entre MVC y MTV .....	67
Tabla 44. Caso de prueba 1 – Alta de un nuevo usuario en la aplicación.....	77
Tabla 45. Caso de prueba 2 – Acceso de un usuario a la aplicación.....	78
Tabla 46. Caso de prueba 3 – Creación de un nuevo análisis.....	79
Tabla 47. Caso de prueba 4 – Lista de resultados.....	79
Tabla 48. Caso de prueba 5 – Validación del formato de ficheros. ....	80
Tabla 49. Descripción de la página de inicio.....	104
Tabla 50. Descripción de la página de alta de nuevos usuarios. ....	105
Tabla 51. Descripción de la página de login. ....	106
Tabla 52. Descripción de la información de asociada a un análisis. ....	107
Tabla 53. Descripción de la información resultante del tipo de análisis CEP. ....	108
Tabla 54. Descripción de la información resultante del tipo de análisis FDA. ....	109

# 1. Introducción

En este primer capítulo se expone la motivación y objetivos que han dado pie al desarrollo de este trabajo, así como la estructura de este documento.

## 1.1. Motivación

El desarrollo de este trabajo fin de grado (TFG) se enmarca dentro del proyecto de investigación: “*Diseño de una metodología de control de parámetros medioambientales e implementación de un sistema de alerta y de protección*” [1]. Dicho proyecto aborda la problemática del desarrollo de una metodología enfocada a la detección de valores atípicos y análisis de tendencias. La metodología está basada en la adaptación de un conjunto de técnicas estadísticas enfocadas al análisis de la calidad ambiental: Control Estadístico de la Calidad (CEC), Control Estadístico de Proceso (CEP) y Análisis Funcional de Datos (AFD) [2] [3].

Lo que se plantea en este TFG es el desarrollo de una aplicación que permita realizar de una forma sencilla una serie de análisis de índole estadístico a un conjunto de datos (valores indicativos de concentración de distintas sustancias: NO, CO<sub>2</sub>, turbidez ...), sobre distintos periodos de tiempo (días, meses o años). Los resultados de dichos análisis serán representados mediante gráficos estadísticos interactivos que ayuden al usuario (experto) en la interpretación de los mismos y en la toma de decisiones.

Para ello se analizará los tipos de datos a registrar, la forma de integrar la herramienta de análisis MAE (*Módulo de Análisis Estadístico*) implementada en Matlab [4], y el tipo de información a representar.

Por tanto, la aplicación web que se desarrolla en este TFG puede aportar un valor añadido a los sistemas de análisis actuales, ya que permitirá complementarlos incrementando la robustez, evitando los falsos positivos por errores de medida y la detección de valores atípicos. De esta manera se busca asistir en la toma de decisiones de estudios ambientales, tanto de la calidad del agua como del aire.

## 1.2. Objetivos

El **objetivo general** de este proyecto es el **desarrollo de una aplicación web** que permita realizar de una manera simple la revisión de los procesos de medida de variables medioambientales. Para conseguirlo, se plantea una serie de **objetivos específicos** que se enumeran a continuación:

- Analizar los entornos existentes para el desarrollo de este sistema software, identificando las tecnologías más adecuadas.
- Poner en práctica los procesos del ciclo de vida del software en un entorno real.
- Diseñar un sistema de validación de la estructura de ficheros con información bruta suministrados por el usuario.
- Integrar la aplicación con un módulo de análisis estadístico implementado en Matlab.
- Diseñar de una interfaz agradable y práctica que permita su utilización tanto en equipos de escritorio como desde dispositivo móviles.
- Crear un sistema de registro de usuarios y autenticación.
- Producir como resultado del análisis, gráficos interactivos que pueden ser descargados por el usuario en su dispositivo.
- Ampliar el alcance de uso de la herramienta.

## 1.3. Estructura del documento

El presente documento se encuentra dividido en siete capítulos, en los cuales se encuentra recogida toda la información relacionada con el desarrollo de este proyecto. A continuación, se muestra un breve resumen del contenido de cada uno de ellos.

### Capítulo 1 – Introducción

En este primer capítulo se establece el contexto en el que se enmarca el desarrollo del proyecto, describiendo los objetivos que se desean alcanzar y su motivación.

### Capítulo 2 – Contexto y estudio preliminar

Una vez llevada a cabo la introducción, se procede a realizar un estudio pormenorizado del dominio de aplicación, identificación de las herramientas existentes y las posibles tecnologías se podrían aplicar.

**Capítulo 3 – Identificación de requisitos**

Partiendo del estudio anterior, en este capítulo se plantean los requisitos identificados para el desarrollo de la aplicación. En este capítulo también se expone la gestión del proyecto o modelo de desarrollo que se va a seguir, la planificación de las distintas fases del proyecto y las herramientas empleadas para el desarrollo.

**Capítulo 4 – Diseño de la solución**

En este capítulo es fundamental debido a que, partiendo de los requisitos, se plasma la arquitectura que tendrá la aplicación, permitiendo tener una descripción a alto nivel de la misma. Para ello se hará uso de distintos tipos de diagramas, modelos, mapas de navegación y prototipos, que orienten y faciliten la fase de implementación.

**Capítulo 5 – Descripción técnica de la solución**

Una vez que se tiene una descripción a alto nivel de cómo debe funcionar la aplicación, y el aspecto que tendrá, este capítulo se centra en explicar la arquitectura empleada, las tecnologías utilizadas, así como los factores técnicos necesarios para la implementación.

**Capítulo 6 – Validación técnica de la solución**

Una vez que la aplicación ya cuenta con una estructura, es necesario valorar si es capaz de responder a las expectativas planteadas. Para ellos se validará el correcto funcionamiento realizando pruebas a cada subsistema, así como a una serie de pruebas de aceptación que permitan saber el grado de satisfacción de los usuarios.

**Capítulo 7 – Conclusiones y líneas de trabajo futuras**

Por último, se recogen las conclusiones extraídas durante el proceso de desarrollo del proyecto, basándose en los objetivos iniciales. Y se plantean las líneas de mejora para el futuro.

## 2. Contexto y estudio preliminar

En este capítulo se detalla el dominio de aplicación, se describen las herramientas existentes (aquellas capaces de realizar algún tipo de análisis estadístico sobre un conjunto de series temporales de datos) y las posibles tecnologías que se pueden emplear para abordar el objetivo de este trabajo.

### 2.1. Dominio de aplicación

Como consecuencia del modelo de desarrollo que actualmente sigue nuestra sociedad, surge un problema ambiental: la contaminación. Cuando se habla de contaminación ambiental se hace referencia a la presencia de ciertos agentes químicos o biológicos en concentraciones que puedan resultar nocivos para la salud humana, la flora y la fauna. Dependiendo del medio que se vea afectado la contaminación puede ser: hídrica, atmosférica o del suelo.

Las emisiones de contaminantes a la atmosfera son la causa actual del deterioro de la calidad del aire. Los elevados niveles de contaminación pueden provocar efectos negativos en la salud humana[5, 6] y en el medio ambiente. Estos niveles, se hacen aún más significativos en grandes núcleos urbanos, o en grandes zonas industriales. Durante la última década (2007-2017) se han dado casos en grandes núcleos urbanos, como es el caso de Madrid [7], Barcelona [8], Londres o París [9], en donde se han tenido que establecer medidas que permitan limitar o reducir el índice de ciertas partículas (o micro partículas) nocivas en el aire. Los contaminantes atmosféricos más frecuentes que se pueden encontrar son: los óxidos de nitrógeno, hidrocarburos, ozono, CO<sub>2</sub>, y los óxidos de azufre, entre otros.

Por otra parte, la calidad del agua puede verse fácilmente alterada por vertidos urbanos, industriales o fertilizantes usados en la agricultura. Se pueden destacar algunos compuestos significativos para evaluar la calidad del agua como son la concentración de amonio y nitratos. Aunque la contaminación de los ríos se expresa mediante un índice global de calidad hecho por los valencianos<sup>1</sup> que se elabora combinando y ponderando los valores de la medición de 23 parámetros.

España cuenta con diversas redes de medición que permiten recoger valores de los indicadores de la calidad, tanto para el agua como es el caso del sistema automático de

---

<sup>1</sup> Índice de calidad general (ICG)

información de calidad de las aguas (SAICA)<sup>2</sup> que cuenta con 200 estaciones fijas, para el aire, con redes de vigilancia de la calidad del aire<sup>3</sup> que cuenta con 600.

El objetivo de estas redes, repartidas por todo el territorio nacional, es producir información continua y transmitirla a diversos ministerios, ayuntamientos, confederaciones hidrográficas, etc., para su posterior explotación.

## 2.2. Herramientas para el análisis estadístico de datos

En esta sección se presentan diversos ejemplos de herramientas que son habitualmente utilizadas en el procesamiento y análisis estadístico de datos. Todas ellas podrían ser utilizadas para procesar datos recogidos en estaciones de medición, y realizar y representar los correspondientes análisis. Por su especial relación con el tipo de aplicación que se desarrolla en este trabajo, son comentadas pormenorizadamente en las siguientes secciones

### 2.2.1. Sistemas de muestreo

Este tipo de sistemas tiene como objetivo principal proporcionar información de diversos parámetros indicativos de contaminación, permitiendo así su detección y evolución en el tiempo, y el análisis de curvas de tendencias. Cabe reseñar la **red de alerta SAICA**, *Sistemas Automático de información de la Calidad*, o las webs de varios ayuntamientos, como es el caso del ayuntamiento de Zaragoza<sup>4</sup>. En estas webs se muestra la evolución temporal de ciertos parámetros considerados indicadores de contaminación, como pueden ser: la cantidad de oxígeno, de amonio, o CO. En la parte superior de la Figura 1, puede verse los últimos valores recibidos de distintos parámetros tales como: temperatura del agua. pH, turbidez, amonio, etc. En la parte inferior aparece la evolución temporal para cada uno de esos valores y en la derecha los correspondientes valores.

---

<sup>2</sup> RED SAICA, disponible en: <http://www.mapama.gob.es/es/agua/temas/estado-y-calidad-de-las-aguas/aguas-superficiales/programas-seguimiento/saica.aspx>

<sup>3</sup> Redes de calidad del aire autonómicas y locales Disponible en: <http://www.mapama.gob.es/es/calidad-y-evaluacion-ambiental/temas/atmosfera-y-calidad-del-aire/calidad-del-aire/evaluacion-datos/redes/default.aspx>

<sup>4</sup> Calidad Aire Zaragoza 2017. Disponible en: [http://www.zaragoza.es/ciudad/medioambiente/atmosfera/redconta/horas\\_CalidadAire](http://www.zaragoza.es/ciudad/medioambiente/atmosfera/redconta/horas_CalidadAire).

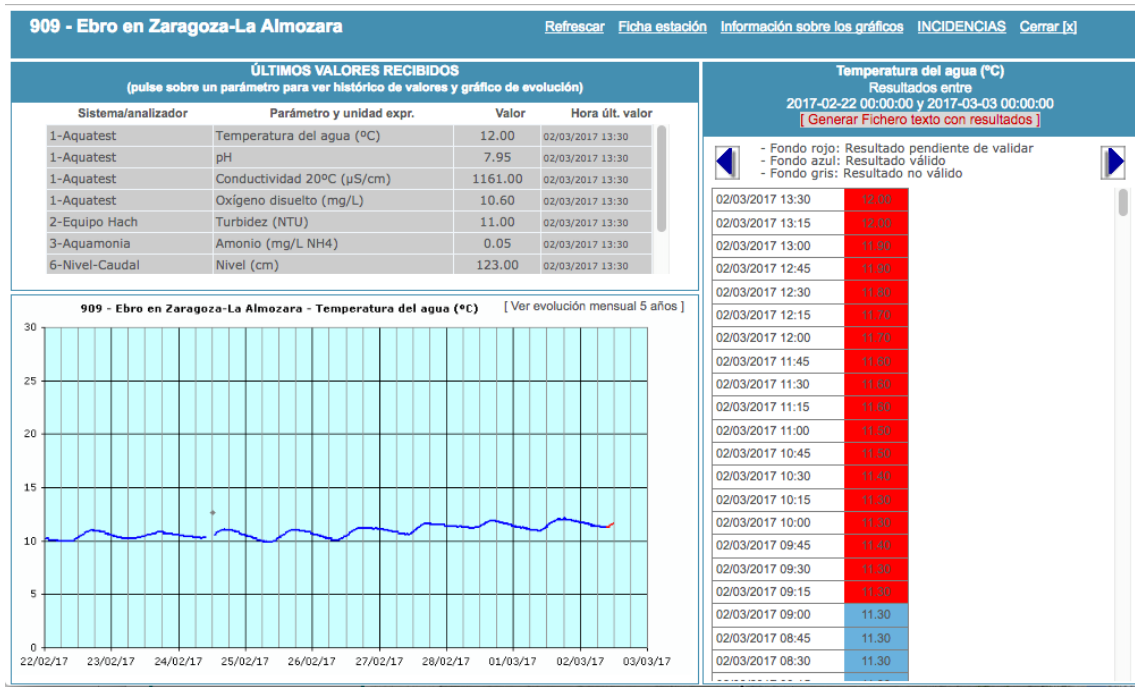


Figura 1. Registros obtenidos por una estación de medición. Red SAICA

En la Figura 2, puede verse la tendencia que ha seguido la concentración de CO en el aire de la ciudad de Zaragoza durante 24 horas.

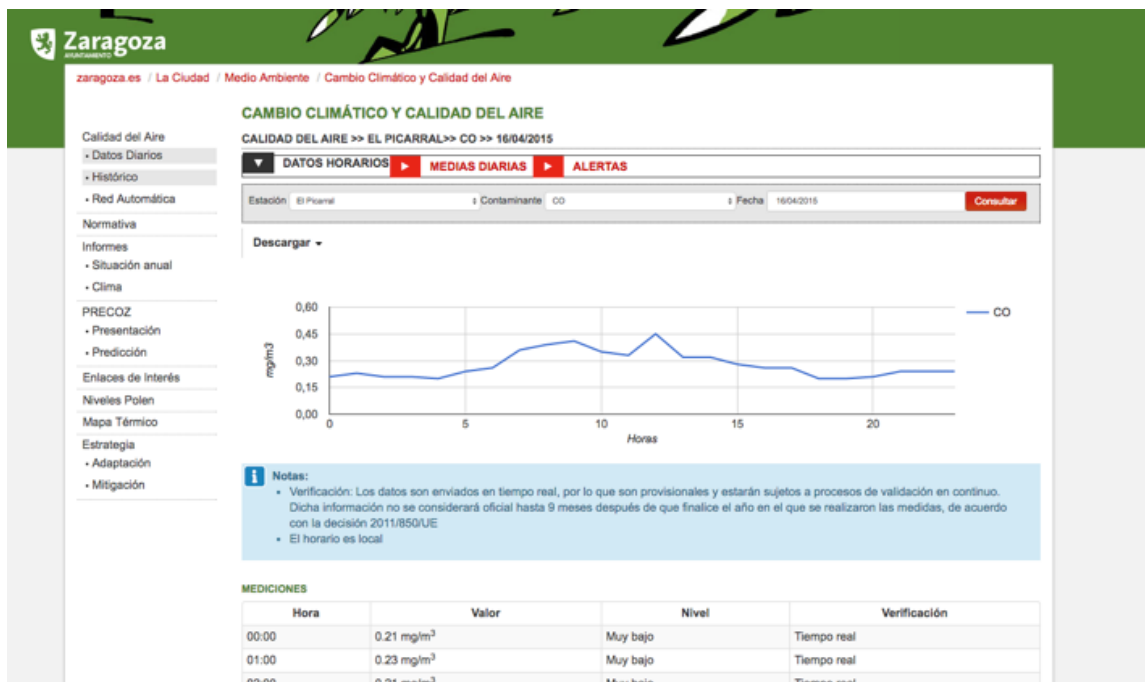


Figura 2. Registros obtenidos por una estación de medición. Ayto ZGZ.

Ambas páginas web realizan un análisis de la serie temporal (curvas de tendencias), pero no son capaces de realizar ningún otro tipo de análisis sobre la información recogida, como por ejemplo análisis de variabilidad. Para poder profundizar más en la información que contienen los datos es necesario el uso de otro tipo de herramientas software como R <sup>5</sup> o Matlab <sup>6</sup>. Ambas cuentan con su propio entorno de programación y su propio lenguaje de desarrollo (R y M), ambas son multiplataforma, R se distribuye bajo licencia GNU GPL mientras que Matlab es de pago. Aunque ambos paquetes de software están muy ampliamente extendidos entre la comunidad investigadora, están muy enfocados a un uso local. Aun así, aunque no cuenten con toda la potencia de una aplicación web, podemos encontrar opciones como las siguientes:

### 2.2.2. R-Markdown <sup>7</sup>

Es un formato de archivo que permite crear documentos dinámicos con R. Básicamente es un archivo escrito en *markdown* en el que se puede integrar código escrito en R, con lo que es posible crear una gran variedad de documentos: html, pdf, Word, odt, rtf, md, etc. En la parte izquierda de la imagen de la Figura 3, pueden verse las distintas secciones del código: definición del tipo de archivo (html), código markdown y código R. En la parte derecha de la imagen se aprecia el documento resultante, en el cual pueden ajustarse ciertos parámetros de la gráfica desde la interfaz web.

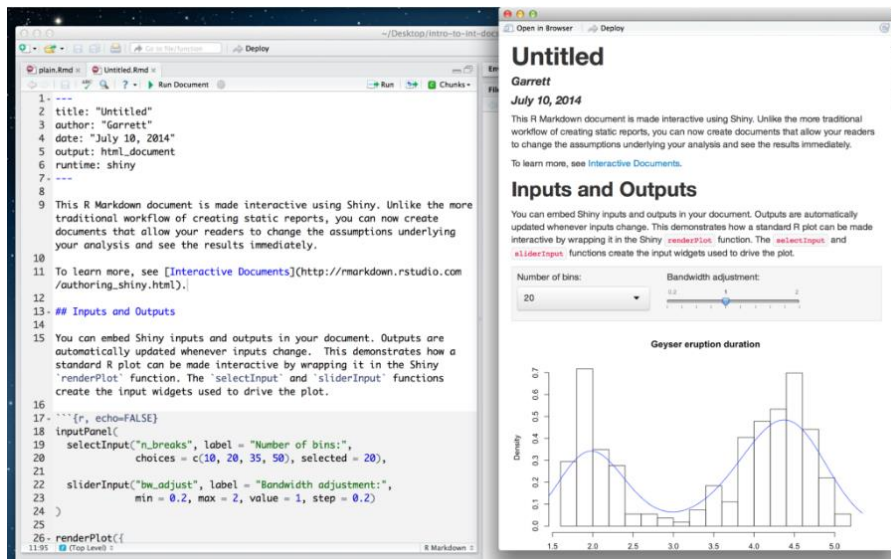


Figura 3. Ejemplo de contenido generado R-Markdown

<sup>5</sup> The R Project for statistical Computing: [www.r-project.org](http://www.r-project.org)

<sup>6</sup> Matlab, página del producto :<http://mathworks.com>

<sup>7</sup> R-Markdown, disponible en: <http://rmarkdown.rstudio.com>

### 2.2.3. Matlab Web Browser

Matlab permite la creación de páginas web interactivas dentro de su propio navegador web, el cual puede interactuar con el espacio de trabajo [10], para realizar los cálculos necesarios que después son mostrados mediante HTML y JavaScript.

Como se puede ver en la Figura 4, la interfaz es capaz de representar la curva de la función seno y permite modificar parámetros de entrada como la amplitud o la frecuencia, así como parámetros de estilo como: color, estilo de línea y marcador.

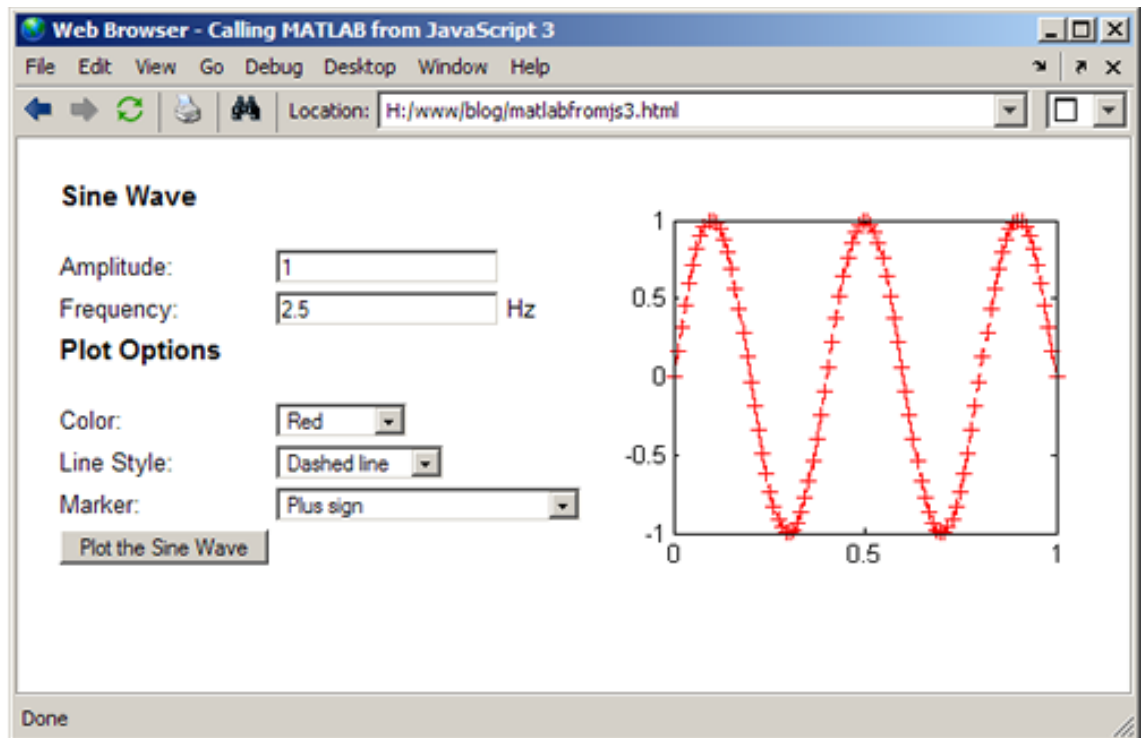


Figura 4. Contenido web generado con Matlab que muestra la curva de la función seno.

El uso de estas herramientas implica una carga extra para el software de cálculo (R o Matlab), porque en caso de modificar los valores, el software tiene que volver a recalcular las funciones y dibujar nuevamente los gráficos.

## 2.3. Herramientas para el desarrollo de aplicaciones web

### 2.3.1. Lenguajes de programación

Será base del proyecto la elección de uno o varios lenguajes de desarrollo, que permitan vertebrar el conjunto de tecnologías a usar para la consecución de los objetivos de este proyecto. Pero esta decisión puede ser complicada, debido al gran número de lenguajes de programación que existen. En la Tabla 1 se muestra una lista de los lenguajes de programación más utilizados.

Marzo 2017	Marzo 2016	Cambio	Lenguaje de programación	Calificaciones	Cambio
1	1		Java	16.384%	-4.14%
2	2		C	7.742%	-6.86%
3	3		C++	5.184%	-1.54%
4	4		C#	4.409%	+0.14%
5	5		Python	3.919%	-0.34%
6	7	+	Visual Basic .NET	3.174%	+0.61%
7	6	-	PHP	3.009%	+0.24%
8	8		JavaScript	2.667%	+0.33%
9	11	+	Delphi/Object Pascal	2.544%	+0.54%
10	14	++	Swift	2.268%	+0.68%
11	9	--	Perl	2.261%	+0.01%
12	10	--	Ruby	2.254%	+0.02%

Tabla 1. Lenguajes de programación más usados en el mundo <sup>8</sup>. + indica una subida de puesto y – indica una bajada de puesto.

Ante la gran variedad de lenguajes de programación, se ha decidido usar Python, y no PHP o Java que sería la opción a la que suele tender ante este tipo de proyectos. La elección de Python como lenguaje principal de desarrollo para este proyecto viene motivada por varias razones: su estabilidad, la claridad de lenguaje, la legibilidad y su enfoque para un mejor mantenimiento, entre otros muchos factores que se presentan a continuación.

**Python** es un lenguaje de alto nivel muy polivalente que permite realizar desde scripting hasta aplicaciones de escritorio o web de gran complejidad [11]. Se trata de un lenguaje interpretado, multiplataforma y multiparadigma. Al tratarse de un lenguaje interpretado, esto le permite una fácil integración con el sistema operativo. Otra de las ventajas de ser un lenguaje interpretado es que permite ciclos de desarrollo breves.

<sup>8</sup> Página de índice Tiobe: <https://www.tiobe.com/tiobe-index/>

Al ser un lenguaje multiparadigma soporta: orientación objetos, programación imperativa y funcional. Permite el tipado dinámico, que ofrece un rico conjunto de estructura de datos flexibles como: listas, cadenas, tuplas, diccionarios etc. Python está muy enfocado a la sintaxis, permitiendo que el código sea limpio y legible. Puede considerarse pseudocódigo ejecutable, ya que es muy expresivo y su sintaxis sencilla interfiere poco en la implementación de algoritmos.

En cuanto al entorno de ejecución, Python facilita la labor de detección y gestión de errores mediante excepciones, además de simplificar la gestión de memoria. Cuenta con una colección de librerías o módulos muy amplia, que permite extender sus funcionalidades y poder trabajar en tareas de tratamiento de datos, visualización, desarrollo de interfaces gráficas, acceso a bases de datos, serialización de objetos, aplicaciones web, etc.

Si a la distribución de software libre en la que se fundamenta, se le suma la comunidad de usuarios y desarrolladores que dan soporte y realizan mejoras, nos encontramos ante un lenguaje muy robusto en continuo desarrollo y evolución.

### **2.3.2. Framework de desarrollo**

Hoy en día contamos con herramientas que facilitan la labor de desarrollo, como es el caso de los *frameworks*. Aunque el termino *framework* se encuentra muy ligado al desarrollo de aplicaciones web, podemos encontrarlos en un buen número de ámbitos del desarrollo software. A grandes rasgos se puede considerar un framework como una aplicación genérica que permite cierto grado de configuración para adaptarla a nuestras necesidades. Los objetivos principales que se buscan con un framework son:

- Acelerar el proceso de desarrollo, simplificando tareas, reutilizando código y reduciendo errores.
- Promover buenas prácticas de desarrollo.
- Establecer una estructura común para el desarrollo de aplicaciones.
- Facilitar la integración de aplicación que utilicen la misma arquitectura.

#### **2.3.2.1. Patrón Modelo Vista Controlador**

Originalmente el patrón Modelo Vista Controlador (MVC) fue concebido para aplicaciones de escritorio, pero se ha adaptado para poder diseñar e implementar aplicaciones web, en casi cualquier lenguaje de programación: .NET; Java, Python, Ruby ...

La popularidad de este patrón entre los frameworks de desarrollo web es debida, a que permite separar los componentes de una aplicación de una manera muy sencilla, permitiendo facilitar y organizar el desarrollo, y el posterior mantenimiento de la aplicación. Este patrón es una guía en diseño de aplicaciones que tienen una gran interactividad con el usuario[12]. El

patrón MVC organiza la aplicación en tres componentes débilmente acoplados: modelo, vista y controlador. En la Figura 5 podemos ver la interacción entre los componentes.

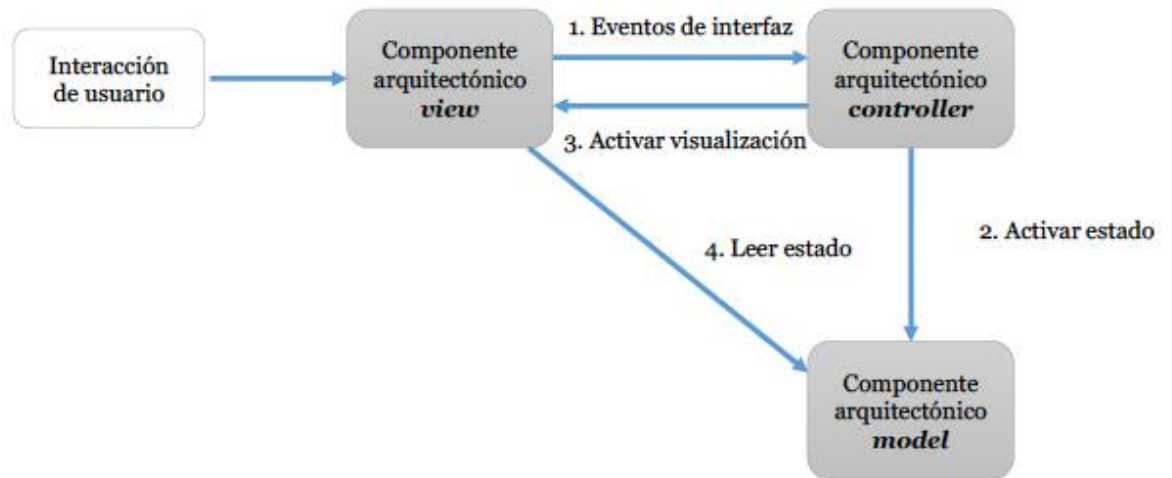


Figura 5. Patrón de interacción de componentes en una arquitectura MVC. Fuente [13].

### Vista

Es el componente encargado de presentar el estado del modelo al usuario. La vista es responsable de dar formato a los datos recibidos del modelo, de una manera accesible para el usuario, ya que los datos pueden venir en diferentes formatos: XML, JSON, etc.

### Modelo

Representa entidades y funcionalidades del núcleo de la aplicación que deben ser manipuladas de forma interactiva por el usuario. El modelo representa los datos y la lógica de una aplicación, lo que se conoce normalmente como **lógica de negocio**. Por lo general se encarga de la gestión de acceso a la información, incluye las operaciones sobre la base de datos, pero tampoco es inusual que la aplicación realice operaciones invocando a servicios web externos o a una API.

### Controlador

Traduce los eventos de interacción de la vista al modelo y selecciona la vista que debe activarse. Como puede verse en la Figura 6, el controlador es el primero, realiza una petición e invoca al modelo; y la respuesta obtenida la envía a la capa de presentación. Es prácticamente el nexo entre el modelo y la vista. Normalmente en el controlador se implementa un constructor, el cual crea una instancia de una clase del modelo y cuando se realiza una solicitud, el controlador decide que datos se precisan del modelo. A continuación, se llama a la clase del modelo para recuperar los datos. Hay que tener en cuenta que el controlador desconoce la base de datos y como se genera la página.

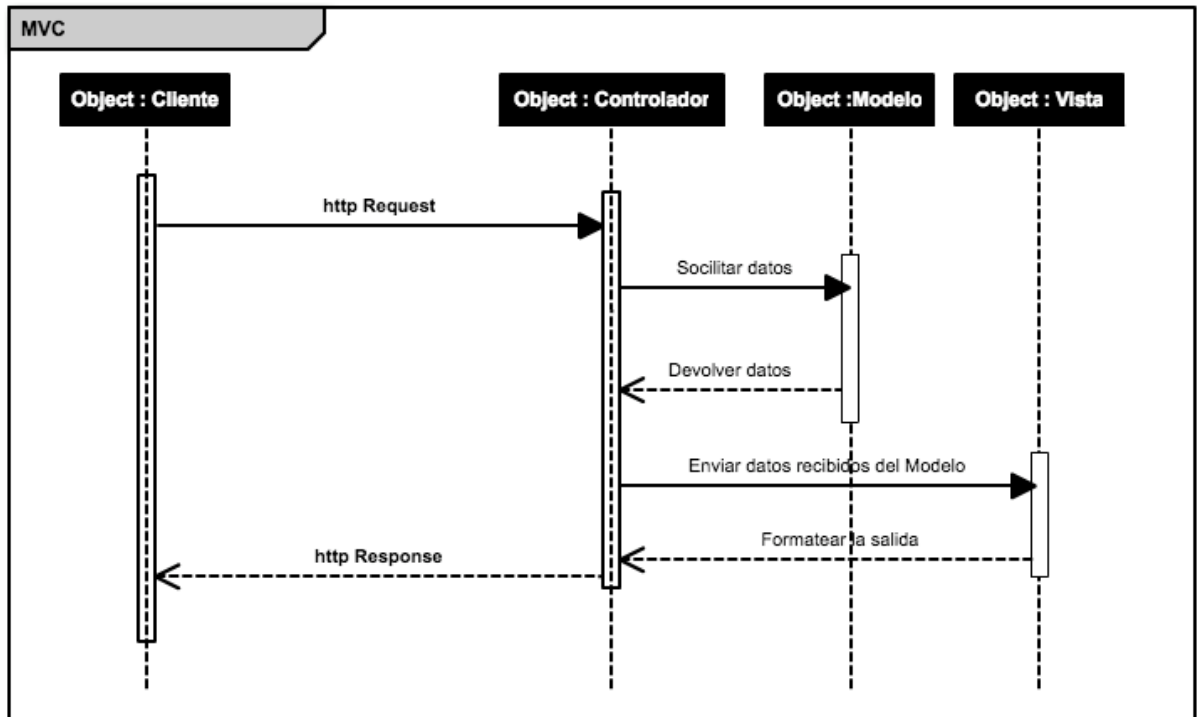


Figura 6. Diagrama de secuencia del patrón MVC donde se puede observar el flujo de control durante una solicitud http.

### 2.3.2.2. Frameworks de desarrollo para aplicaciones web

Este tipo de frameworks están enfocados al desarrollo web. Mediante su uso se pretende reducir la carga de trabajo asociada a las actividades que implica un desarrollo web, mediante un diseño reutilizable que facilite y agilice el desarrollo. Algunos ejemplos son: Django para Python, Rails para Ruby o Symphony para PHP entre otros. Este tipo de frameworks se caracterizan por contar con:

- Sistema web de plantillas.
- Sistema de cache web, que ayude a reducir la carga del servidor y el uso de ancho de banda.
- Sistemas de autenticación y autorización, que permiten al servidor identificar a los usuarios de la aplicación y restringir el acceso.
- Mapeo, configuración y acceso a bases de datos; permitiendo abstraer a la aplicación de la base de datos y el uso de mapeo objeto-relacional.
- Mapeo de URL, que permite el uso de url más amigables y cortas.
- Internacionalización y localización.

A la hora de llevar a cabo desarrollos web utilizando Python, nos encontramos una serie de frameworks que destacan sobre el resto, como son: **Django** y **Flask**. Como puede verse en la

Figura 7, estos no son los únicos frameworks, aunque si los más usados según el ranking establecido en *HotFrameworks*.

Django y Flask comparten una serie de premisas como es el de favorecer la productividad, contando con herramientas que permiten desarrollar en el menor tiempo posible. Aunque también tienen diferencias o han sido creados para distintos propósitos, como se muestra en los siguientes apartados.

### Django

Django <sup>9</sup>, es un framework de código abierto de gran envergadura, para desarrollo rápido, siguiendo su lema “*dont repite your self*” [14]. El enfoque de Django “*batteries included*”, incluye todo lo que un desarrollador pueda necesitar, por lo que solo es necesario instalarlo, empezar a trabajar, e ir usando los módulos de Django a medida que sean necesarios.

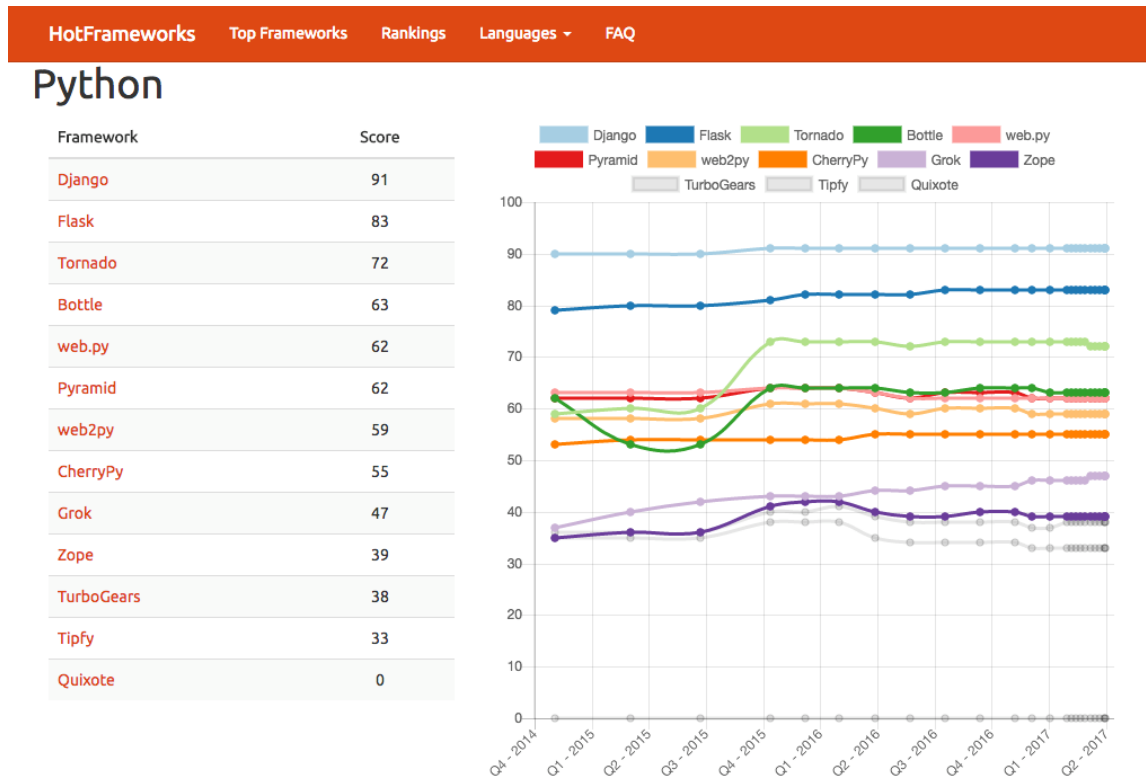


Figura 7. Ranking basado en la popularidad de los desarrolladores. Imagen obtenida de la página Hotframework <sup>10</sup>.

<sup>9</sup>Página de Django Project: <https://www.djangoproject.com/>

<sup>10</sup> Hotframework, disponible en: <https://hotframeworks.com/>

La estructura del proyecto es autogenerada, además de contar con un sistema de administración que simplifica la gestión de tareas más comunes, como el sistema de internalización, la gestión de la base de datos, etc.

Se trata de un framework verdaderamente maduro, y que ha acumulado un gran número de plugins y extensiones para poder satisfacer una amplia gama de necesidades. Y, además, también cuenta con una muy buena documentación y multitud de ejemplos.

Aunque también plantea una serie de cuestiones que no son tan favorables, ya que se trata de un proyecto bastante cerrado, la curva de aprendizaje es muy pronunciada, lo que puede llevar a que sea muy costoso empezar un proyecto, y la gran cantidad de características puede llegar a ser abrumadora.

Django presenta una orientación a la realización de proyectos grandes y en equipo. Es práctico para páginas o aplicaciones que precisen cierto grado de interacción con el usuario.

### Flask

Flask <sup>11</sup>, se trata de un micro-framework web para Python, sencillo, rápido y flexible. Es simple y extensible, y puede añadirse complejidad a medida que sea necesario. Está enfocado principalmente a pequeñas aplicaciones con requisitos más simples y cuenta con una documentación clara y práctica.

Por contra, el desarrollo al principio es rápido, pero a medida que se amplían las necesidades puede ir aumentando gradualmente la complejidad y habrá que hacer uso de herramientas de terceros desarrolladores.

Mediante la siguiente Tabla 2 podemos comparar algunas de las características más reseñables de ambos frameworks:

	<b>DJANGO</b>	<b>FLASK</b>
<b>CATEGORIA</b>	Framework	Micro-framework
<b>BOOTSTRAPING<sup>12</sup></b>	SI (django-admin)	NO
<b>ORM</b>	PROPIO	<i>SQLAlchemy*</i>
<b>ALCANCE POR YECTO</b>	MEDIO/GRANDE	PEQUEÑO
<b>CARACTERISICAS</b>	<i>batteries included<sup>13</sup></i>	FLEXIBILIDAD/SIMPLICIDAD
<b>CURVA DE APRENDIZAJE</b>	ALTA	BAJA
<b>SISTEMA DE PLANTILLAS</b>	PROPIO	<i>JINJA2*</i>

<sup>11</sup> Flask, página del proyecto: <http://flask.pocoo.org/>

<sup>12</sup> Bootstrapping: herramientas que facilitan el arranque del proyecto

<sup>13</sup> Batteries included: resuelve gran parte de los problemas, ya que incluye un gran número de librerías/módulos

<b>I18N INTERNACIONALIZACION</b>	PROPIO	<i>BABEL</i> *
<b>SIGNAL</b>	PROPIO	<i>BLINKER</i> *
<b>MODELO MVC</b>	SI	NO

Tabla 2. Tabla comparativa de características entre Django y Flask.

### 2.3.3. Base de datos

En casi cualquier aplicación web es necesario que la información resida sobre alguna estructura lógica, y pueda ser accesible. Para este fin se pueden usar las bases de datos. Una **base de datos** [15] es una colección de información que está organizada para permitir que sea accesible, administrada y actualiza de una manera más sencilla. Los datos que conforman la base de datos se encuentran organizadas y pueden evolucionar a medida que se va modificando (añadiendo o quitando) la información.

#### 2.3.3.1. Sistema gestor de base de datos

Para gestionar una base de datos existe un tipo de software denominado **Sistema Gestor de Bases de Datos** o SGBD que permite: definir, manipular, recuperar y gestionar los datos almacenados en una base de datos. Para las bases de datos relaciones se pueden encontrar los siguientes software gratuitos: *Sqlite, Mysql, PostgreSQL*; y de pago: *Access, Oracle, y Sqlserver*. A continuación, en la Tabla 3, se van a presentar las características más destacadas de estos tres SGBDR que distribuyen de manera gratuita.

	<b>Mysql</b>	<b>PostgreSQL</b>	<b>SQLite</b>
<b>Modelo</b>	Relacional	Objeto-Relacional	Relacional
<b>Versión</b>	5.7.18	9.6.2	3.18.0
<b>Licencia</b>	GPL V2	BSD	Public Domain
<b>OS</b>	Multiplataforma	Multiplataforma	Server-less
<b>Arq. cliente-servidor</b>	SI	SI	NO
<b>SQL</b>	Con extensiones propietarias	SQL-92	SQL-92 no soportado entero
<b>Integridad referencial</b>	SI	SI	SI
<b>ACID</b>	SI (InnoDB)	SI	SI
<b>Concurrencia</b>	SI (InnoDB)	SI (MVCC)	SI (bloqueo de archivos del sistema)

Tabla 3. Comparativa de características: SQLite, Mysql y PostgreSQL

#### 2.3.3.2. Object-Relational Mapper

*Object-relational mapper* (ORM) [16] es una librería que permite automatizar la transferencia de datos almacenados en tablas de una base de datos relacional a objetos que

se utilizan comúnmente en las aplicaciones. En la Figura 8, se muestra a modo de ejemplo una posible relación entre los campos de un objeto en Python y los campos de una tabla. ORM proporciona un nivel de abstracción sobre una base de datos relacional, lo que permite a los desarrolladores trabajar de manera más cómoda en un mismo lenguaje de programación, en lugar de tener que usar consultas SQL: para crear, leer, actualizar y eliminar datos y esquemas de bases de datos.

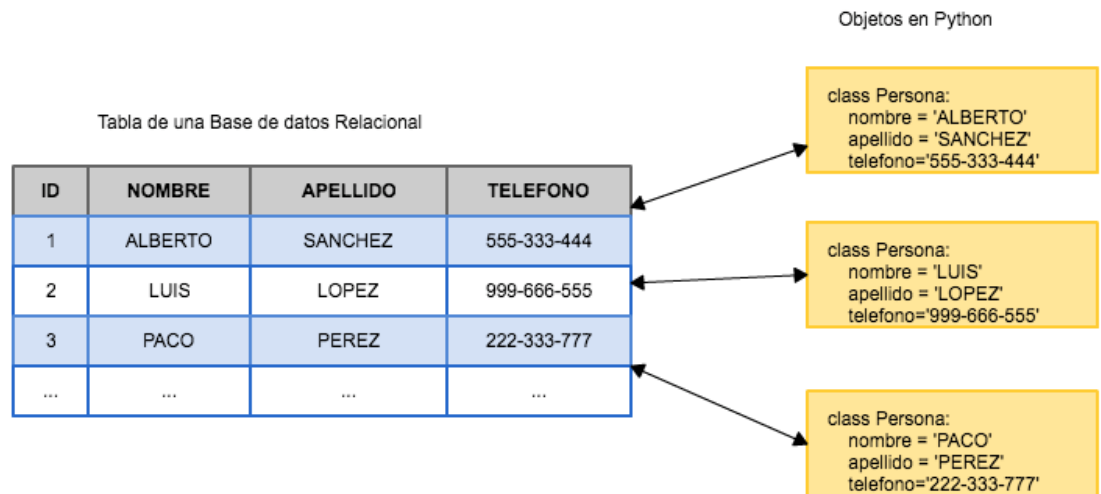


Figura 8. ORM provee de un “puente” entre las tablas, relaciones, campos de las bases de datos relacionales y los objetos Python.

Las sentencias en ORM se denominan **queryset** (QS) y son básicamente una lista de objetos de un modelo determinado. Mediante el QS podemos leer datos de una base de datos, filtrarlos y ordenarlos. A modo de ejemplo se pueden ver las siguiente comparativa entre una sentencia SQL para recuperar a la información del usuario con id=1

***Select \* from users where id=1***

Y la consulta equivalente en ORM sería:

***Usuarios = Users.objects.filter(id=1)***

Las características más destacables del uso de un ORM son:

- **Encapsulación:** dentro del objeto ORM está el código SQL que hace conexiones con la base de datos.
- **Reutilización de código:** permitiendo llamar al mismo método desde varias partes de la aplicación sin reescribir código.

- **Abstracción:** permitiendo cambiar de base de datos sin apenas realizar ningún cambio en el código.
- **Creación de métodos:** que devuelvan estructuras de datos como *strings*, con los datos solicitudes a la base de datos.
- **Velocidad de desarrollo:** sobre todo al comienzo, al no tener que estar alternando entre lenguajes, como Python o PHP y SQL.

En cuanto a las **desventajas o inconvenientes** de utilizar un ORM:

- **La complejidad de la base de datos reside en la aplicación:** con ORM el código de manipulación de datos en lugar de residir en la base de datos, está dentro del código de la aplicación, incrementan la cantidad total de código. Por tanto, en vez de separar el código de la aplicación de los procedimientos almacenados de la base de datos todo reside en el mismo sitio
- **Object-Relational Impedance mismatch** [17]: se trata del problema que existe en ORM, ya que la forma en la se trabaja con los objetos difiere de cómo se almacenan y se unen en las tablas de una base de datos relacional
- **Perdida de rendimiento:** la perdida de rendimiento puede venir asociada a la traducción que se produce del QS a una instrucción SQL.

#### 2.3.4. Framework CSS

El desarrollar la interfaz de nuestra aplicación web desde cero puede ser una tarea costosa. Para facilitar esta labor de desarrollo, existen frameworks que permiten simplificar, agilizar y acelerar el proceso de desarrollo. Un **framework-CSS** [18] también denominado **framework frontend**, es básicamente un paquete que está compuesto de archivos HTML, CSS y *JavaScript* ya predefinidos, que pueden emplearse para empezar a construir un proyecto. En un **framework-CSS** por lo general nos vamos a encontrar los siguientes componentes:

- Código **HTML**, que ayuda a formar la estructura de las páginas.
- Estilos de **tipografías**.
- **CSS** para cambiar visualmente como aparecen los elementos.
- **JavaScript** para cambiar elementos dinámicos: como menús desplegables, ...

- *Responsive media queries*(\*)<sup>14,15</sup>
- Compatibilidad entre navegadores.

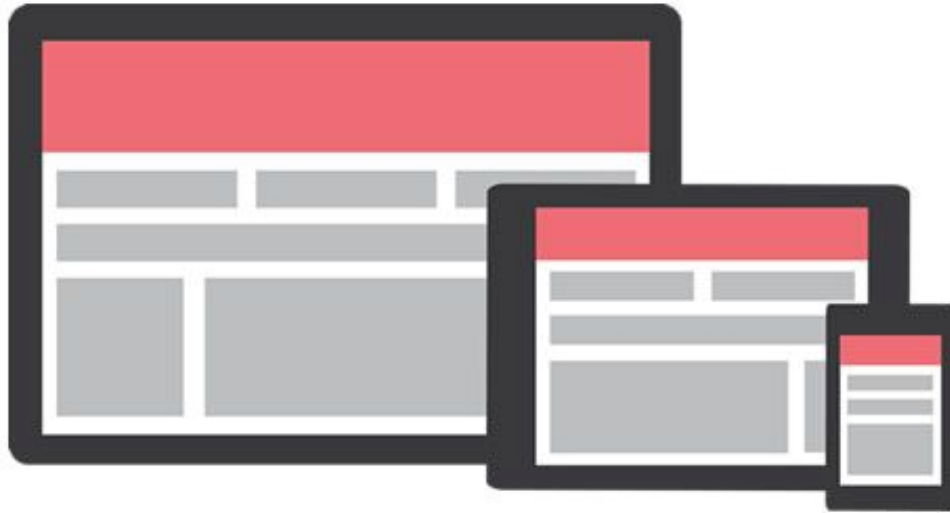


Figura 9. La imagen muestra la funcionalidad responsive web design, que permite poder adaptar el contenido de una página al dispositivo que lo está visualización. Imagen obtenida de la página Materializecss<sup>16</sup>

### Ventajas y desventaja del uso de un framework CSS

Son muchos los puntos a favor que tiene el uso de un framework CSS, siendo el más destacable probablemente el de no tener que iniciar cada proyecto web completamente desde cero, aunque también podemos destacar los siguientes:

- Es más fácil mantener el código, usando *snippets*<sup>17</sup> y librerías.
- Está más organizado y es más fácil de configurar.
- *Responsive media queries*(\*) y compatibilidad con navegadores.
- La mayoría son gratuitos y de código abierto.
- Es estable y está bien testeado por cientos de desarrolladores.
- Es posible contribuir en el proyecto mediante Github.

<sup>14</sup> **Responsive web design** hace referencia al uso de diferentes técnicas que permiten adaptar el diseño de un sitio web a las características del medio como: ancho, alto y color.

<sup>15</sup> **Media queries**: se trata de un módulo CSS3 que permiten identificar el medio en el cual se está mostrando el sitio y adaptar la representación del contenido a las características del dispositivo. Son el elemento principal del **responsive web design**.

<sup>16</sup>Web del proyecto Materializecss : <http://materializecss.com>

<sup>17</sup>**Snippets**: término usado en programación para referirse a pequeñas partes reusables de código.

- Se pueden obtener actualizaciones periódicas, corrección de errores y nuevas funcionalidades.
- Se pueden usar para prototipado: producir wireframes y prototipos para acelerar el proceso de diseño y desarrollo.

En contrapartida se pueden destacar ciertas desventajas o inconvenientes del uso de este tipo de framework:

- Al trabajar con un código ya prediseñado, pueden darse ciertas limitaciones en cuanto a la elección del diseño o tener algún problema si quiere corregir algún efecto no deseado.
- Al importar código del framework que no es necesario para el diseño web puede provocar ciertos incrementos en los tiempos de carga de la pagina.
- 

Debido a su popularidad contamos con una amplia gama de frameworks frontend como: *Bootstrap*, *Semantic-UI*, *Foundation*, *Blueprint*, etc. En la Tabla 4, se ha realizado una breve comparativa de los dos frameworks más usados: **Bootstrap**<sup>18</sup> [25] y **Foundation**<sup>19</sup>.

		<b>Bootstrap</b>	<b>Foundation</b>
<b>VERSION</b>		3.3.7	6
<b>LICENCIA</b>		MIT	MIT
<b>CSS PREPROCESOR</b>		LESS/SAAS	SAAS
<b>GRID</b>	<b>LAYOUT</b>	12 COLUMNAS	12 COLUMNAS
	<b>CONFIGURACION</b>	<i>FLUID / FIXED</i>	<i>FLUID / FIXED / CENTERING</i>
	<b>ANCHO</b>	1170PX	940PX
<b>UNIDADES</b>		PIXEL	REMS
<b>SOPORTE NAVEGADORES</b>		Chrome Safari FireFox Opera IE8+	Chrome Safari FireFox Opera IE9+
<b>VALIDACION FORMULARIO</b>		3THPARTY	ABIDE
<b>TAMAÑO ARCHIVO</b>		370KB	206KB

Tabla 4. Comparativa de las características más destacables de Bootstrap y Foundation

Ambos framework son muy potentes y pueden ayudar a agilizar el proceso de diseño. Aunque Bootstrap es más popular que Foundation, no hay realmente un framework mejor que otro,

<sup>18</sup> Bootstrap: <http://getbootstrap.com/>

<sup>19</sup> Foundation: <http://foundation.zurb.com>

así que la elección se reduce a la experiencia que se tenga en desarrollo, con cual se está más cómodo, y el que permita lograr la apariencia que se quiera, con el menor esfuerzo

### 2.3.5. Visualización de datos

La capacidad de obtener información, de un conjunto de datos en bruto, está sujeta a la capacidad que tenga uno mismo para transformarla, para que esos datos tengan un significado [20]. El entender los datos como elementos aislado, estos no serán capaces de aportar ninguna información concreta. Es solo, al aplicar sobre ellos una interpretación, cuando toman sentido y por lo tanto pasan a ser conocimiento. Hoy en día existen mecanismos cada vez más robusto para la explotación de datos, siendo la visualización de datos uno de los más importante.

La **visualización de datos** es la presentación gráfica de la información, y cuyos objetivos son: construir un significado a partir de los datos y la comunicación. Es decir, permite descubrir y entender la lógica que se encuentra en un conjunto de datos, y compartir, desde un punto de vista objetivo, una misma interpretación con otras personas.

La visualización de datos, está basada en unos principios formales, que permite diseñar representaciones graficas las cuales expresan mejor un significado de manera gráfica, que verbal.

#### 2.3.5.1. Estándares web para visualización de datos

El uso de tecnologías basadas en estándares permite que los navegadores web proporcionen los gráficos de manera nativa sin el uso de ningún tipo de complemento. Es por ello que en este apartado se van describir los estándares: HTML5<canvas>, SVG y WebGL, empleados en la creación de visualizaciones en entornos web.

#### Estándar HTML5 Canvas

HTML5 Canvas<sup>20</sup> se trata de una API *JavaScript* [29] para la elaboración de gráficos vectoriales sobre un mapa de bits de un tamaño específico. Canvas se presenta como un sistema de gráficos que trabaja en **modo inmediato**, donde se expone una superficie en la cual crear y manipular imágenes rasterizada pixel a pixel. Una vez pintada el sistema se olvida de ellas, es decir olvida el contexto en el que se representó.

---

<sup>20</sup> Canvas API: [https://developer.mozilla.org/en-US/docs/Web/API/Canvas\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API).

Esto plantea una serie de ventajas:

- Reduce la memoria necesaria para mantener un modelo interno (DOM).
- Se pueden aprovechar los métodos y propiedades de API de HTML5 para referenciar al objeto dentro del contexto 2D. Por ejemplo, 10px desde la izquierda del canvas.
- Permite dibujar de manera más rápida sin necesidad de retener la información.
- Permite leer y escribir píxeles.
- Se ajusta mejor a la representación de datos en tiempo real.

Aunque también presenta ciertas limitaciones:

- Los eventos y funciones a capturar son limitados, dado que no hay un conocimiento retenido del gráfico.
- Dependiendo de la resolución las imágenes creadas en el CANVAS, pueden perder calidad cuando se agranda o se muestran en pantallas de gran resolución
- La ventana del explorador no puede ajustarse de forma nativa
- El movimiento de un objeto por el canvas implica volver a dibujar toda la escena
- El explorador no puede convertir de forma nativa las coordenadas de la pantalla al sistema de coordenadas del mundo

### **Estándar SVG**

Es el acrónimo de *Scalable Vector Graphics (SVG)*<sup>21</sup> o gráficos vectoriales escalables, que de acuerdo con su especificación [29]:

*“SVG es un lenguaje para describir gráficos bidimensionales. Es un formato independiente o puede usarse con XML, HTML... Los gráficos SVG pueden ser interactivos y dinámicos. Las animaciones se pueden definir y activar de forma declarativa (es decir, mediante la incorporación de elementos de animación) o por medio de secuencia de comandos”*

SVG presenta un enfoque de trabajo a más alto nivel, en donde cada figura dibujada es recordada como un objeto en un escenario gráfico o DOM, que posteriormente es pasado a un mapa de bits. Así es como trabaja el **modo retenido**, en donde los gráficos persisten en un modelo de memoria que se puede manipular mediante código.

---

<sup>21</sup> Web del proyecto SVG: <https://svgwg.org/svg2-draft/intro.html>

Como puntos a destacar se puede identificar que:

- Cada objeto que se dibuja, añade un modelo interno, lo que facilita ciertas tareas para el desarrollo, pero con un cierto coste para el rendimiento.
- Al ser escalable tiene la ventaja, al contrario que las imágenes generadas por HTML5-canvas, es posible aumentar o reducir una imagen vectorial manteniendo su calidad.
- Puede ser tratado como un archivo independiente o puede ser integrado en HTML junto con el resto de elementos, atributos y estilos.
- No es necesario rehacer toda escena, cada vez que se mueve un objeto.
- Proporciona un matriz de transformación para convertir coordenadas de pantalla a coordenadas de representación de gráficos.

Aunque también presenta ciertas limitaciones:

- Cuantos más objetos hay presentes en el DOM más puede disminuir el rendimiento.
- Puede penalizar en uso de recursos, para mantener el modelo interno.

A continuación, en la Tabla 5, se expone una comparativa de las características más destacables de ambos sistemas:

	<b>CANVAS</b>	<b>SVG</b>
<b>Estado o modo de representación</b>	Inmediato	Retenido
<b>Dependiente de la resolución</b>	SI	NO
<b>Trabaja a nivel de</b>	Pixel	DOM
<b>Interacción de eventos</b>	Granular	Abstraída
<b>Mejor rendimiento en superficies</b>	Pequeña	Grande
<b>Numero objetos manipular</b>	> 10K	< 10 K
<b>Manipulación mediante</b>	Script	Script/xml/css
<b>Características claves</b>	Velocidad/Sencillez	Interactividad/Flexibilidad/ Accesibilidad/Calidad de imagen
<b>Peso de imágenes (bytes)</b>	Mayor	Menor
<b>Uso de memoria</b>	Menor	Mayor

Tabla 5. Tabla comparativa de las características más destacables de Canvas frente a SVG. ¡Error! Marcador no definido.

Además de la característica clave como es el uso del **modo de representación inmediato o retenido**, también son muy significativas las características de **tamaño de la pantalla o número de objetos a representar**, que tienen un impacto muy significativo en el tiempo de representación de imágenes. Aunque es cierto que se presentan escenarios donde es más

apropiado el uso de una tecnología que la otra, como se puede ver en el estudio comparativo a alto nivel de varios escenarios para SVG y Canvas [22], cabe decir que es posible conseguir resultados casi idénticos haciendo uso de ambas tecnologías.

### **Estándar WebGL**

*Web Graphics Library*<sup>22</sup> es una API JavaScript para representar gráficos interactivos en 2D y en 3D dentro de cualquier navegador web compatible [29] sin el uso de complementos o *plugins* mediante una API que hace uso de *OpenGL ES 2.0*. Mediante esta API, WebGL permite el renderizado 3D sobre un HTML<canvas> en los navegadores que lo soporten. Los programas WebGL consisten en: código de control escrito en *JavaScript* y un código de efectos especiales (*shader code*) que se ejecuta en la GPU (unidad de procesamiento gráfico). Los elementos WebGL se puede mezclar con otros elementos HTML y combinar con otras partes de la página web. Permite distintos niveles de abstracción, desde el más bajo que sería prácticamente implementar en OpenGL hasta el uso de distinto frameworks como Three.js o x3D. También cabe destacar WebGL 2 que es una actualización importante de WebGL que está basado en *OpenGL ES 3.0* e incluye una serie de nuevas características como: texturas 3D, sincronización de objetos, etc.

#### **2.3.5.2. Librerías para gráficos 2D**

Podemos encontrar un gran número de librerías que nos permiten visualizar nuestros datos mediante gráficos 2D o incluso 3D. En la Tabla 6 se muestra una breve comparativa de las librerías que se van revisar. La primera columna (Estándar) indica el estándar, de los descritos anteriormente, sobre el que se apoya su implementación. La segunda (paleta de gráficos) indica la percepción obtenida del análisis/estudio de la librería, pudiendo ser: escasa, si los gráficos son pocos y no pueden modificarse, amplia, si cuenta con un gran número de gráficos ya predefinidos, o extensible, que además de contar con un gran número de gráficos ya prediseñados permite, mediante las primitivas de dibujo, la realización de cualquier gráfico que se necesite. En cuanto a la tercera columna (Uso), todas son de uso gratuito, aunque Highcharts y Plot.ly cuentan con versiones de pago para entornos más comerciales. La última columna (Curva de aprendizaje) presenta la percepción obtenida de haber realizado ciertas pruebas de desarrollo con estas librerías.

---

<sup>22</sup> WebGL 2.0 disponible en :<https://www.khronos.org/registry/webgl/specs/latest/2.0/>.

	Estándar	Paleta de gráficos	Uso	Curva aprendizaje
<b>D3.js</b>	SVG	Extensible	Gratis	Alta
<b>Google Charts</b>	SVG	Amplia	Gratis	Baja
<b>Chart.js</b>	CANVAS	Escasa	Gratis	Baja
<b>Highcharts</b>	SVG	Amplia	Gratis*	Media
<b>Plot.ly</b>	SVG/WebGL	Extensible	Gratis*	Media

\*cuentan con versiones comerciales de pago

Tabla 6. Librerías para gráficos 2D

### D3.js

*Data Driven Documents*[23] también conocida como D3. se trata de una librería *JavaScript* orientada a la manipulación de documentos basados en datos, y que se sustenta en los estándares web HTML, SVG y CSS. Mediante D3 se pueden llevar a cabo gráficos interactivos y personalizados, aplicables a cualquier ámbito y de gran complejidad. Para ello expone una capa de abstracción sobre SVG, mediante una serie de utilidades que permiten la manipulación de los datos y su visualización. Cuenta con un amplísimo repositorio de ejemplos, ya predefinidos (Figura 10) o compartidos por la comunidad<sup>23</sup> o, si se necesita, se pueden crear desde cero haciendo uso de primitivas de dibujo, como puntos, líneas, etc.



Figura 10. Ejemplo de la variedad de la paleta de gráficos que pueden implementarse con D3. Imagen obtenida de: <https://github.com/d3/d3/wiki/Gallery>.

Además de ser una de las librerías más utilizadas para la creación de visualizaciones, también sirve de base para otras librerías más específicas como NVD3, C3js, etc. que permiten reducir la complejidad que presenta la utilización de D3. Para ello estas librerías

<sup>23</sup> Galeria de ejemplos D3: <https://bl.ocks.org/mbostock>.

tienen preparadas las clases JavaScript de los principales tipos de gráficas, gráficos de barras, de líneas, de área, etc., que facilitan la creación de visualización.

### Google Chart

Se trata de una librería desarrollada por Google [24]enfocada a la visualización de datos en sitios web. Esta librería está basada en SVG, y cuenta con un gran número de tipos de gráficos ya predefinidos, que van desde los más simples, como gráficos de líneas, a otros con mayor complejidad como mapas de árbol jerárquico. Los gráficos son básicamente clases JavaScript, que permiten un cierto grado de personalización e interactividad.

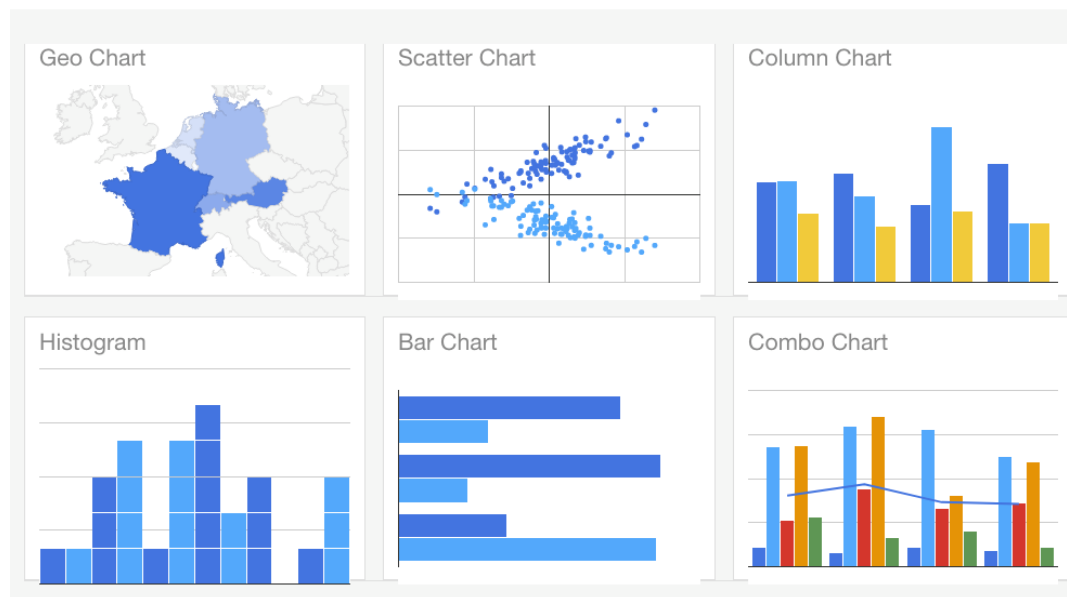


Figura 11. Ejemplo de los gráficos a usar con Google Charts. Imagen obtenida de:  
<https://developers.google.com/chart/interactive/docs/gallery>

Todos los gráficos se nutren de datos, a través de un protocolo denominado *Chart tools DataSource*, que es un lenguaje de consultas muy similar a SQL y que permite obtener datos almacenados en Google Spreadsheets, *Google Fusión Tables* y proveedores de datos de terceros como *SalesForce*. Indicar que también cuenta con ciertas limitaciones en cuanto al número de consulta por día y en cuanto al tamaño de la gráfica.

## Chart.js

Es una de las librerías [25] más rápidas y fáciles de aprender a usar, ya que destaca por su simplicidad y facilidad para personaliza. Se trata de una librería *JavaScript* que permite dibujar distintos tipos de gráficos haciendo uso del elemento HTML5 Canvas. El usar Canvas en lugar de SVG, por una parte limita en cierta medida la interoperabilidad de los gráficos con CSS, pero presenta la ventaja de tener que usar menor recursos para su representación. La librería no cuenta con dependencias y ocupa/pesa realmente poco 11kb (minified). Otra característica destacable es que los *charts* son *responsive*, es decir, que se pueden adaptar en función del espacio disponible. En contrapunto, cabe señalar la escasa variedad de gráficos, ya que solo cuenta con 8 tipos predefinidos.

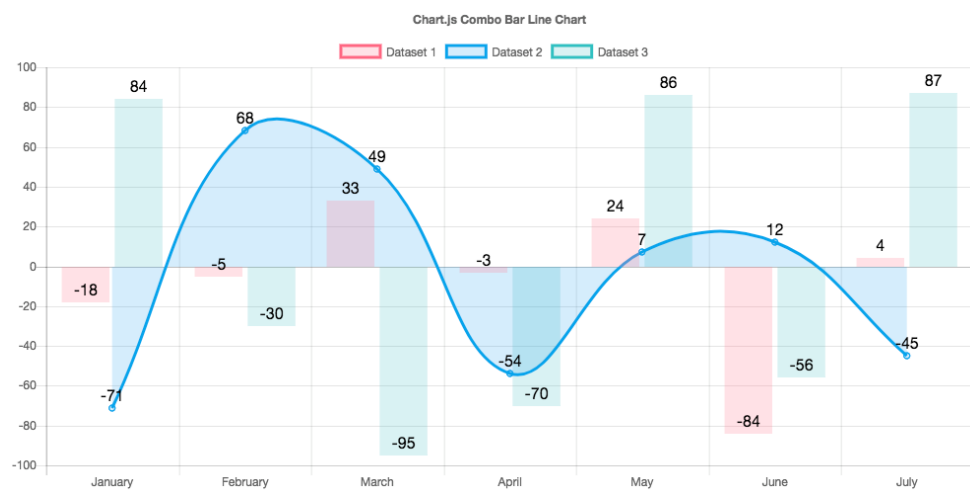


Figura 12. Ejemplo de gráfico creado con Chart.js. Fuente <sup>24</sup>.

## Highcharts

Esta librería JavaScript[26] también hace uso del estándar SVG para crear los gráficos. La configuración e implementación es fácil ya que ofrece una sintaxis de configuración simple. Destaca por el amplio soporte de graficas de todo tipo que le permiten abordar la mayor parte de casuísticas que pueden darse. Cuenta también con un editor que permite crear gráficos de una manera sencilla, API para distintos lenguajes: .NET, IOs, Java, etc.

<sup>24</sup> <http://www.chartjs.org/samples/latest/charts/combo-bar-line.html>

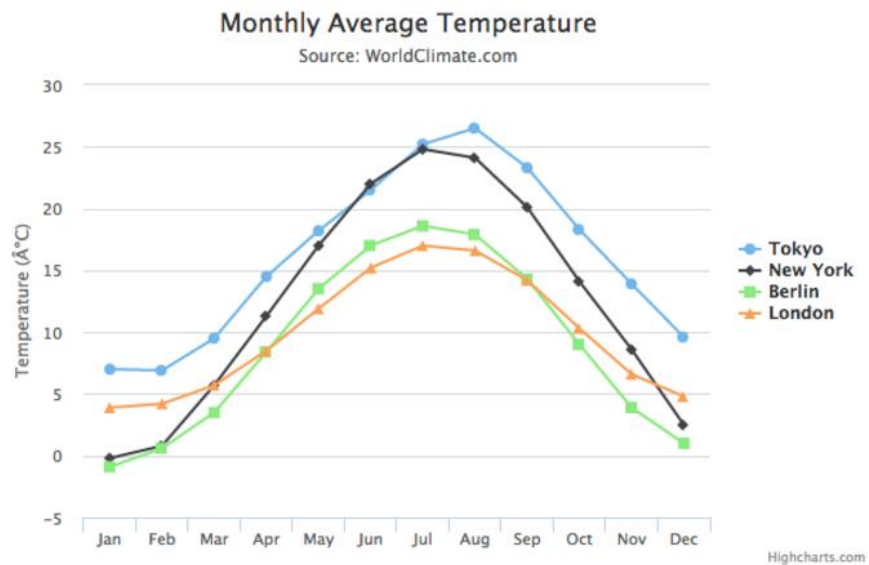


Figura 13. Ejemplo de grafico creado con Highcharts. Fuente<sup>25</sup>

Aunque se trata de una librería comercial, presenta distintos tipos de usos, entre ellos el no comercial. En la cual se puede hacer uso de casi todas las la graficas de la plataforma. Además, permite una gran flexibilidad para adaptarla a plataforma requerida, editarla e incluso combinar múltiples gráficos en uno.

### Plot.ly

Plotly[27] es una librería de gráficos *JavaScript*, que permite crear y compartir visualizaciones de datos; y a su vez ofrece herramientas de análisis estadístico. Cuenta con una gran comunidad de desarrolladores trabajando tanto en el *core* de la librería como en los clientes para los distintos lenguajes (R, Python, Matlab, etc), además de contar con un gran número de ejemplos y repositorios compartidos por la comunidad.

Plotly puede definirse como un paquete todo en uno, ya que hace uso de D3.js(SVG) y de módulos stack.gl (WebGL) para la representación gráfica, por lo que no tiene dependencias externas. A esto también se suma que han eliminado el uso de *jquery*, lo que ha supuesto una mejora significativa en el rendimiento, así como una mejor compatibilidad con los navegadores. WebGL le permite la representación interactiva de cientos de miles millones de puntos xy, mientras que D3.js es más práctico para un máximo de decenas de miles de puntos, así como exportar imágenes vectoriales de gran calidad.

<sup>25</sup> <https://www.highcharts.com/demo/line-labels>

Plotly soporta más de 20 tipos de gráficos, incluyendo: Gráficos 3D, Mapas geográficos, Estadísticos, de densidad, Histogramas, Diagramas de cajas o Gráficos de contorno entre muchos otros. Además, permite la creación de cualquier tipo de gráfico altamente personalizable, ya que cuenta con un editor que facilita esta tarea.

#### Statistical and Seaborn-style Charts [↗](#)

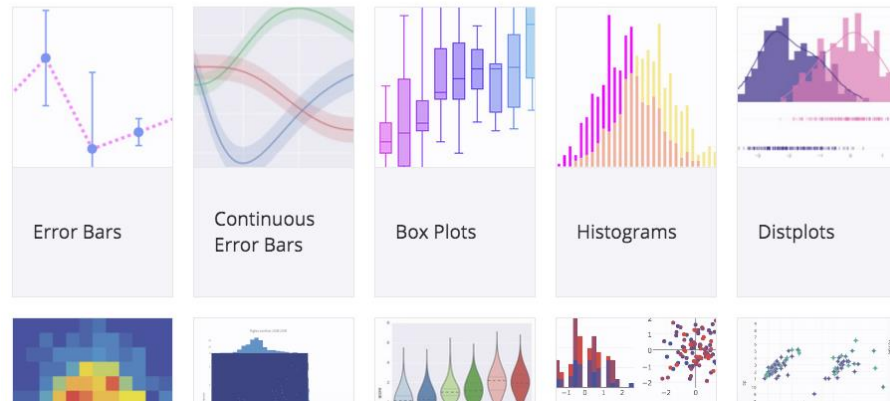


Figura 14. Clasificación de la paleta de gráficos soportados por Plotly.

## 3. Identificación de requisitos

A partir del estudio anterior, y teniendo en cuenta las características particulares de la aplicación a desarrollar, en este capítulo se va a describir los requisitos necesarios para poder llevar a cabo del desarrollo de la aplicación. También se va a describir la metodología seguida para la gestión del proyecto, así como las herramientas empleadas para el desarrollo del proyecto.

### 3.1. Especificación de requisitos software

Los requisitos software son el documento donde se plasma por escrito la naturaleza exacta de la aplicación. Se elaboran a partir de los requisitos del usuario y son la base para el diseño y la implementación. La especificación de requisitos software ERS [28] es una descripción del comportamiento del sistema que se pretende desarrollar.

La ERS contiene **requisitos funcionales (RQF)** y **requisitos no funcionales (RQNF)**. Los primeros, derivados de los requisitos de capacidad, especifican la funcionalidad o servicios que la aplicación debe proporcionar. Además, incluye también un conjunto de casos de uso (CDU), en los que se describen todas las interacciones que tendrá los usuarios con el software. En cuanto a los segundos, los RNF, son derivados de los requisitos de restricción, los cuales imponen restricciones en el producto desarrollado, o en el proceso de desarrollo. Los RNF responden al *cómo* debe realizar algo el software, y no *qué* debe realizar.

#### 3.1.1. Requisitos funcionales

A continuación, se exponen los requisitos funcionales identificados, siguiendo la siguiente estructura:

- Código identificativo
- Nombre descriptivo
- Versión y fecha de creación o de modificación
- Autor
- Tipo (funcional/no funcional),
- Una descripción pormenorizada del almacenase del mismo
- Prioridad, catalogada como: baja, media o alta

<b>Código</b>	<b>RQF - 1</b>
<b>Nombre</b>	Crear un nuevo análisis
<b>Versión</b>	1 (27/5/2017)
<b>Autor</b>	Alberto
<b>Tipo</b>	Funcional
<b>Descripción</b>	El sistema deberá poder crear un nuevo análisis a partir de la información aportada por el usuario. La información introducida por el usuario será validada antes de proceder a la creación de un proceso de cálculo. Se validará que los campos estén completados, que las fechas seleccionadas tengan una lógica propia del tipo de análisis y la información contenida en el fichero proporcionada en el fichero.
<b>Prioridad</b>	Alta

Tabla 7. RQF 1 – Crear nuevo análisis.

<b>Código</b>	<b>RQF - 2</b>
<b>Nombre</b>	Selección del tipo de análisis
<b>Versión</b>	1 (27/5/2017)
<b>Autor</b>	Alberto
<b>Tipo</b>	Funcional
<b>Descripción</b>	El sistema solo permitirá seleccionar los siguientes tipos de análisis a realizar: CEP si se quiere realizar un análisis de control estadístico de procesos. FDA si se quiere realizar un análisis funcional de datos.
<b>Prioridad</b>	Alta

Tabla 8. RQF 2 – Selección del tipo de análisis.

<b>Código</b>	<b>RQF - 3</b>
<b>Nombre</b>	Periodo temporal sobre el que realizar el análisis
<b>Versión</b>	1 (27/5/2017)
<b>Autor</b>	Alberto
<b>Tipo</b>	Funcional
<b>Descripción</b>	El sistema permitirá seleccionar una de los siguientes periodos: día, mes o año.
<b>Prioridad</b>	Alta

Tabla 9. RQF 3 – Selección del periodo temporal.

<b>Código</b>	<b>RQF - 4</b>
<b>Nombre</b>	Intervalo temporal sobre que acotar el análisis.
<b>Versión</b>	1 (27/5/2017)
<b>Autor</b>	Alberto
<b>Tipo</b>	Funcional
<b>Descripción</b>	El sistema solicitará un intervalo temporal sobre el que realizar el análisis.
<b>Prioridad</b>	Alta

Tabla 10. RQF 4 – Fichero con información a analizar.

<b>Código</b>	<b>RQF - 5</b>
<b>Nombre</b>	Archivo con la información en bruto a analizar
<b>Versión</b>	1 (27/5/2017)
<b>Autor</b>	Alberto
<b>Tipo</b>	Funcional
<b>Descripción</b>	El sistema solicitará un archivo que contendrá la información a analizar.
<b>Prioridad</b>	Alta

Tabla 11. RQF 5 – Fichero con información a analizar.

<b>Código</b>	<b>RQF - 6</b>
<b>Nombre</b>	Validar del fichero proporcionado por el usuario
<b>Versión</b>	1 (27/5/2017)
<b>Autor</b>	Alberto
<b>Tipo</b>	Funcional
<b>Descripción</b>	El sistema permitirá comprobar si el fichero que el usuario quiere analizar cumple con las características de formato indicadas. Formato (xls), estructura del fichero y el tipo (fecha y valor) de los campos.
<b>Prioridad</b>	Alta

Tabla 12. RQF 6 – Validar fichero.

<b>Código</b>	<b>RQF - 7</b>
<b>Nombre</b>	Borrar un análisis seleccionado
<b>Versión</b>	1 (27/5/2017)
<b>Autor</b>	Alberto
<b>Tipo</b>	Funcional
<b>Descripción</b>	El sistema permitirá borrar toda la información relacionada con el análisis seleccionado. La información creada durante el análisis y los resultados obtenidos del mismo. Solo podrá borrar la información que haya sido creada y asociada a dicho usuario. Se pedirá confirmación al usuario antes de borrar la información.
<b>Prioridad</b>	Alta

Tabla 13. RQF 7 – Borrar análisis

<b>Código</b>	<b>RQF - 8</b>
<b>Nombre</b>	Listar todos los análisis creados por el usuario.
<b>Versión</b>	1 (27/5/2017)
<b>Autor</b>	Alberto
<b>Tipo</b>	Funcional
<b>Descripción</b>	El sistema permitirá mostrar en un listado los análisis creados por el usuario, en caso de no haber ningún se indicará que no hay todavía. La información mostrar será: título, tipo de análisis, fecha de creación y el estado del proceso.
<b>Prioridad</b>	Alta

Tabla 14. RQF 8 – Listar análisis creados.

<b>Código</b>	<b>RQF - 9</b>
<b>Nombre</b>	Paginar lista de resultados.
<b>Versión</b>	1 (27/5/2017)
<b>Autor</b>	Alberto
<b>Tipo</b>	Funcional
<b>Descripción</b>	La lista de resultados será paginada en función del número de elementos que haya en la lista.
<b>Prioridad</b>	Baja

Tabla 15. RQF 9 – Paginar la lista de resultados.

<b>Código</b>	<b>RQF - 10</b>
<b>Nombre</b>	Filtrar los resultados en función del campo seleccionado por el usuario.
<b>Versión</b>	1 (27/5/2017)
<b>Autor</b>	Alberto
<b>Tipo</b>	Funcional
<b>Descripción</b>	Se permitirá agrupar los resultados en función de uno de los campos indicados en la lista.
<b>Prioridad</b>	Baja

Tabla 16. RQF 10 – Filtrar la lista de resultados.

<b>Código</b>	<b>RQF - 11</b>
<b>Nombre</b>	Indicar el estado en el que se encuentra el proceso de análisis.
<b>Versión</b>	1 (27/5/2017)
<b>Autor</b>	Alberto
<b>Tipo</b>	Funcional
<b>Descripción</b>	El sistema indicara el estado en el que se encuentra un proceso de análisis, pudiendo ser: pendiente de procesar, procesando, calculando finalizado o error.
<b>Prioridad</b>	Alta

Tabla 17. RQF 11 – Indicar el estado del proceso de análisis.

<b>Código</b>	<b>RQF - 12</b>
<b>Nombre</b>	Acceder al contenido de un análisis.
<b>Versión</b>	1 (27/5/2017)
<b>Autor</b>	Alberto
<b>Tipo</b>	Funcional
<b>Descripción</b>	El sistema permitirá el acceso a la información del análisis, y cuando estén disponibles a los resultados para ese análisis.
<b>Prioridad</b>	Alta

Tabla 18. RQF 12 – Acceder a información análisis.

<b>Código</b>	<b>RQF - 13</b>
<b>Nombre</b>	Actualizar la información de un análisis seleccionado.
<b>Versión</b>	1 (27/5/2017)
<b>Autor</b>	Alberto
<b>Tipo</b>	Funcional
<b>Descripción</b>	El sistema permitirá modificar los campos: título y comentarios del análisis indicado, así como los ejes de las gráficas, correspondientes al tipo de análisis.
<b>Prioridad</b>	Media

Tabla 19. RQF 13 – Actualizar la información de un análisis.

<b>Código</b>	<b>RQF - 14</b>
<b>Nombre</b>	Resultado del tipo de análisis: Control Estadístico de procesos (CEP).
<b>Versión</b>	1 (27/5/2017)
<b>Autor</b>	Alberto
<b>Tipo</b>	Funcional
<b>Descripción</b>	Se mostrará la información resultante del proceso de análisis CEP mediante gráficas y datos.
<b>Prioridad</b>	Alta

Tabla 20. RQF 14 – Resultado de análisis CEP.

<b>Código</b>	<b>RQF - 15</b>
<b>Nombre</b>	Resultado del tipo de análisis: Análisis Funcional de datos (FDA).
<b>Versión</b>	1 (27/5/2017)
<b>Autor</b>	Alberto
<b>Tipo</b>	Funcional
<b>Descripción</b>	Se mostrará la información resultante del proceso de análisis: Análisis Funcional de Datos (AFD) mediante gráficas y datos.
<b>Prioridad</b>	Alta

Tabla 21. RQF 15 – Resultado de análisis FDA.

<b>Código</b>	<b>RQF - 16</b>
<b>Nombre</b>	Descarga de resultados.
<b>Versión</b>	1
<b>Autor</b>	Alberto
<b>Tipo</b>	Funcional
<b>Descripción</b>	Los resultados obtenidos podrán descargarse mediante: PNG o PDF.
<b>Prioridad</b>	Baja

Tabla 22. RQF 16 – Descargar los resultados generados.

### Diagrama de casos de uso: Sistema de análisis

Mediante el siguiente Diagrama de casos de uso, véase Figura 15, se procede a describir la interacción entre los **actores**, es este caso, **usuarios autenticados** y el **sistema**, en referencia a los requisitos funcionales RQF del 1 al 12.

**Descripción:**

Un usuario autenticado podrá realizar cuatro tareas principales.

1. Crear un nuevo análisis. (RQF – 01).
2. Validar un fichero, para comprobar que cumple con el formato requerido (RQF – 02)
3. Listar los análisis que haya creado. (RQF – 04, 05, 06, 07).
  - i. En caso de existir alguna instancia / análisis creado previamente, el sistema indicará el estado del proceso.
 

Una vez que el análisis haya concluido el usuario podrá acceder a la información generada en función de tipo de análisis que haya seleccionado: CEP o FDA. (RQF – 08,09,10,11,12).
  - ii. En caso de no haber creado ningún hasta el momento se le indicara que no existen.
  - iii. Se podrá filtrar los resultados en función del campo seleccionado.
4. Borrar un análisis, y toda la información asociada. (RQF – 03).

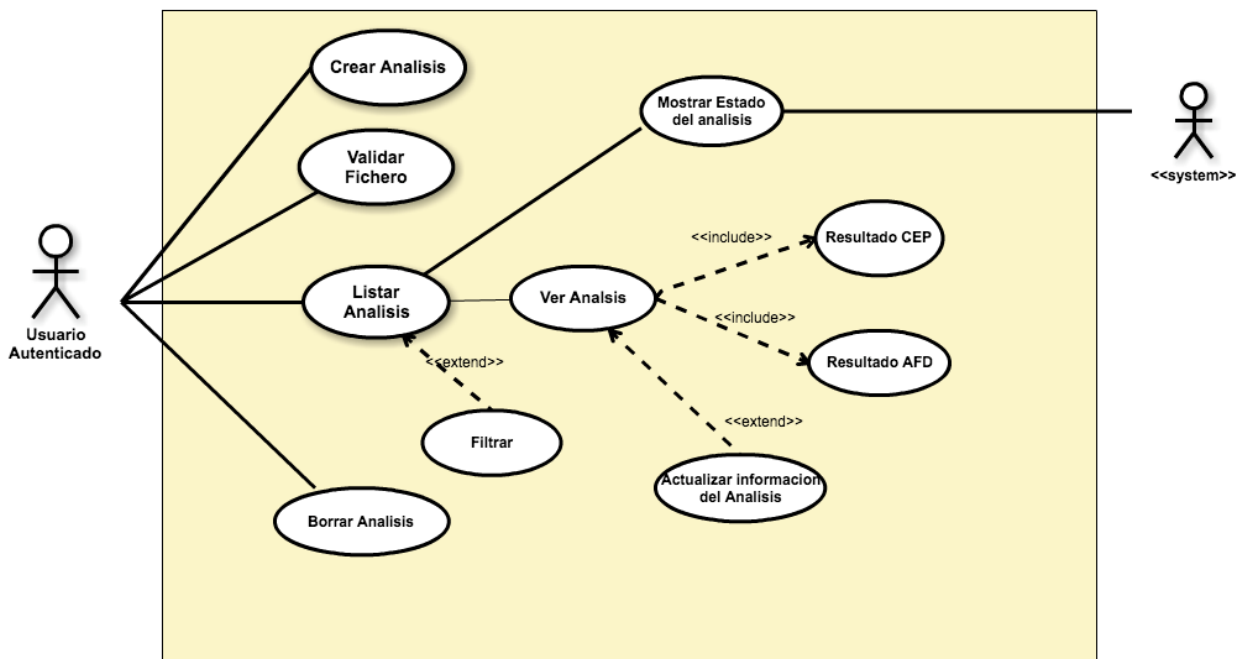


Figura 15. Diagrama de caso de uso: Sistema de Análisis.

<b>Código</b>	<b>RQF - 17</b>
<b>Nombre</b>	Sistema de alta de nuevos usuarios.
<b>Versión</b>	1
<b>Autor</b>	Alberto
<b>Tipo</b>	Funcional
<b>Descripción</b>	El sistema permitirá que los usuarios puedan registrarse en el sistema, proporcionando su nombre, correo electrónico y contraseña. Una vez realizado el proceso de registro se les enviará un correo que les permita activar la cuenta.
<b>Prioridad</b>	Alta

Tabla 23. RQF 17 – Registro de un nuevo usuario.

<b>Código</b>	<b>RQF - 18</b>
<b>Nombre</b>	Activación de la cuenta de usuario
<b>Versión</b>	1
<b>Autor</b>	Alberto
<b>Tipo</b>	Funcional
<b>Descripción</b>	El sistema enviará un correo con un enlace que permitirá a los usuarios activar su cuenta.
<b>Prioridad</b>	Alta

Tabla 24. RQF 18 – Activación de la cuenta de usuario.

<b>Código</b>	<b>RQF - 19</b>
<b>Nombre</b>	Sistema de autenticación.
<b>Versión</b>	1
<b>Autor</b>	Alberto
<b>Tipo</b>	Funcional
<b>Descripción</b>	Si el usuario ya tiene creada una cuenta podrá acceder a su información, haciendo uso de su usuario y contraseña.
<b>Prioridad</b>	Alta

Tabla 25. RQF 19 – Sistema de autenticación.

<b>Código</b>	<b>RQF - 20</b>
<b>Nombre</b>	Restaurar contraseña de la cuenta de un usuario.
<b>Versión</b>	1
<b>Autor</b>	Alberto
<b>Tipo</b>	Funcional
<b>Descripción</b>	El sistema permitirá al usuario restaurar la contraseña en caso de que se haya olvidado, enviándole un enlace por correo electrónico que permita generar una nueva contraseña.
<b>Prioridad</b>	Media

Tabla 26. RQF 20 – Restaurar contraseña.

<b>Código</b>	<b>RQF - 21</b>
<b>Nombre</b>	Acceso a la información de cada usuario.
<b>Versión</b>	1
<b>Autor</b>	Alberto
<b>Tipo</b>	Funcional
<b>Descripción</b>	Cada usuario solo podrá acceder al contenido que haya creado: análisis, resultados y perfil.
<b>Prioridad</b>	Alta

Tabla 27. RQF 21 – Perfil y contenido de usuario.

<b>Código</b>	<b>RQF - 22</b>
<b>Nombre</b>	Formulario de contacto.
<b>Versión</b>	1
<b>Autor</b>	Alberto
<b>Tipo</b>	Funcional
<b>Descripción</b>	El sistema contará con un formulario de contacto, que permita dirigirse, tanto a los usuarios registrados como a los no registrados (anónimos), al administrador de la aplicación.
<b>Prioridad</b>	Media

Tabla 28. RQF 22 – Formulario de contacto

<b>Código</b>	<b>RQF - 23</b>
<b>Nombre</b>	Panel de Staff
<b>Versión</b>	1
<b>Autor</b>	Alberto
<b>Tipo</b>	Funcional
<b>Descripción</b>	Los usuarios que pertenezcan al staff, podrán acceder a información de: estadísticas de uso, los archivo reportados por los usuarios ...
<b>Prioridad</b>	Media

Tabla 29. RQF 23 – Panel Staff.

### **Caso de uso: Sistema de registro y autenticación de usuarios**

Mediante el siguiente Diagrama de casos de uso, véase Figura 16, se procede a describir la interacción entre los **actores**, es este caso, **usuarios no autenticados, autenticados y staff**, con el **sistema**, en referencia a los requisitos funcionales RQF del 13 al 19.

#### **Descripción:**

1. Un usuario que no tenga una cuenta creada en el sistema, tiene la posibilidad de darse de alta en el sistema, mediante un registro y la posterior activación de la cuenta.
2. En caso de tener una cuenta, el usuario hará uso de sus credenciales (usuario y contraseña) para acceder al sistema.
  - i. En caso de no recordar la contraseña se le enviará un nueva.
  - ii. Una vez que esté dentro del sistema el usuario podrá acceder a toda la información que está asociada a su perfil.
3. Tanto los usuarios autenticados como los no autenticados tienen a su disposición la opción de contactar con el administrador.
4. Los usuarios que pertenezcan al grupo Staff contarán con un panel que les permita tener información global del sistema.

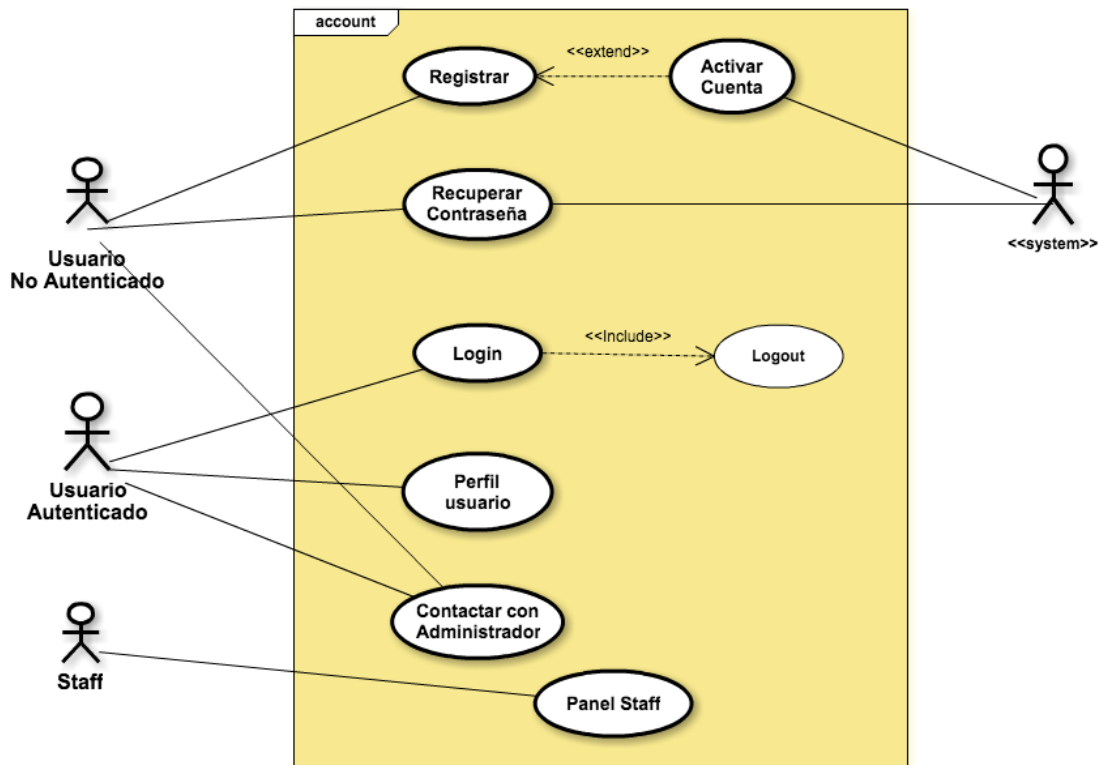


Figura 16. Diagrama de Caso de uso: Sistema de registro y autenticación de usuarios.

### 3.1.2. Requisitos No funcionales

<b>Código</b>	<b>RQNF - 1</b>
Nombre	Comunicación con software de Análisis Matlab.
Versión	1 (27/5/2017)
Autor	Alberto
Tipo	Funcional
Descripción	El sistema tiene que poder comunicarse con el software de análisis MATLAB mediante su API <sup>26</sup> .
Prioridad	Alta

Tabla 30. RQNF 1 – Comunicación con MATLAB.

<sup>26</sup> MATLAB API para otros lenguajes: <https://es.mathworks.com/help/matlab/programming-interfaces-for-c-c-fortran-com.html>

<b>Código</b>	<b>RQNF - 2</b>
Nombre	Cada usuario solo tendrá acceso a su información.
Versión	1 (27/5/2017)
Autor	Alberto
Tipo	Funcional
Descripción	Cada usuario solo podrá acceder a aquella información que haya sido generada por el mismo. Es decir, todos los análisis y resultados de los mismos quedaran asociados a él, así como su información de perfil.
Prioridad	Alta

Tabla 31. RQNF 2 – Información asociada a cada usuario.

<b>Código</b>	<b>RQNF - 3</b>
Nombre	Diseño de la interfaz <i>responsive</i> .
Versión	1 (27/5/2017)
Autor	Alberto
Tipo	Funcional
Descripción	La interfaz de la aplicación tiene que poder adaptarse a la resolución de la pantalla del dispositivo desde el que se accede a la aplicación.
Prioridad	Alta

Tabla 32. RQNF 3 – Diseño de interfaz adaptativa.

<b>Código</b>	<b>RQNF - 4</b>
Nombre	Validaciones sobre los campos de fechas del formulario.
Versión	1 (27/5/2017)
Autor	Alberto
Tipo	Funcional
Descripción	Se impondrán una serie de restricciones sobre las fechas, que el usuario debe complementar en el formulario de creación de un nuevo análisis.
Prioridad	Alta

Tabla 33. RQNF 4 – Validaciones de los campos de los formularios.

<b>Código</b>	<b>RQNF - 5</b>
Nombre	Validaciones sobre el fichero proporcionado por el usuario.
Versión	1 (27/5/2017)
Autor	Alberto
Tipo	Funcional
Descripción	Se llevarán a cabo una serie de comprobaciones sobre el formato y estructura del fichero que proporciona el usuario.
Prioridad	Alta

Tabla 34. RQNF 5 – Validaciones sobre ficheros.

<b>Código</b>	<b>RQNF - 6</b>
Nombre	Uso de <i>datetimepicker</i> para introducción de fecha y hora
Versión	1 (27/5/2017)
Autor	Alberto
Tipo	
Descripción	En los campos en los que sea necesario introducir algún valor temporal se hará uso de la herramienta <i>datetimepicker</i> ..
Prioridad	Alta

Tabla 35. RQNF 6 – Datetimepricker.

### 3.2. Gestión del proyecto

En este apartado se va a exponer la metodología de desarrollo software a emplear durante el ciclo de vida de este proyecto, así como una propuesta de la planificación a seguir, identificando cada una la fase correspondiente al proyecto.

#### 3.2.1. Ciclo de vida

El ciclo de vida de un proyecto se corresponde con una metodología o modelo de desarrollo software. Una metodología software consta de un conjunto de actividades enfocadas a la creación de un producto software [29].

Aunque existe un buen número de metodologías de desarrollo, y cada una tiene sus peculiaridades, todas se asemejan o comparten etapas o fases fundamentales en la elaboración de cualquier proyecto, como son:

- **Especificación:** donde se definen las funcionalidades y restricciones.
- **Diseño e implementación:** desarrollo de un software que cumpla con las especificaciones realizadas anteriormente.
- **Validación:** se muestra el software al cliente para que dé su conformidad.
- **Evolución:** se corresponde a los cambios que hay que realizar para ajustar el software a los requisitos del cliente.

Para la consecución de este proyecto se va a plantear el uso del **modelo de desarrollo iterativo incremental**, que se apoya en cierta medida en el **modelo de desarrollo en cascada**.

#### 3.2.2. Modelo de desarrollo en cascada

El desarrollo en cascada es un enfoque metodológico que ordena rigurosamente las etapas del proceso de desarrollo software [30], de tal manera que cada etapa no comienza hasta la anterior haya finalizado. Al finalizar cada etapa, se lleva a cabo una revisión que

permite determinar si es posible pasar a la siguiente fase. Se trata de un modelo fácil de implementar y de entender, orientado a documentos y que promueve una metodología de trabajo efectiva: *definir antes que diseñar, diseñar antes que codificar*.

Las fases de las que se compone este modelo son la siguientes:

- **Análisis de requisitos:** en esta fase se trabaja con el usuario para analizar las necesidades y determinar los objetivos a cumplir. Es una fase muy importante ya que una vez especificados los requisitos es muy difícil modificarlos en las siguientes fases del proyecto.
- **Diseño del sistema:** se organiza la estructura que conformará el sistema, identificando los diferentes módulos y las relaciones entre ellos.
- **Implementación:** en esta fase se procede a la codificación/programación del sistema.
- **Pruebas:** se procede a ensamblar los diferentes módulos o partes que se han desarrollado y que conforman el sistema para comprobar su correcto funcionamiento y validar el cumplimiento de los requisitos establecidos al principio.
- **Verificación:** consta del despliegue o implantación del sistema en producción y la aceptación final del usuario.
- **Mantenimiento:** una vez entregado al cliente, el software puede sufrir cambios motivados por errores encontrados, ampliación de funcionalidades, o mejoras de rendimiento.

En la Figura 17 se puede ver un ejemplo de la disposición de las fases que componen el desarrollo en cascada. Los errores detectados en la fase de pruebas implican un rediseño y programación, aumentando el coste de desarrollo

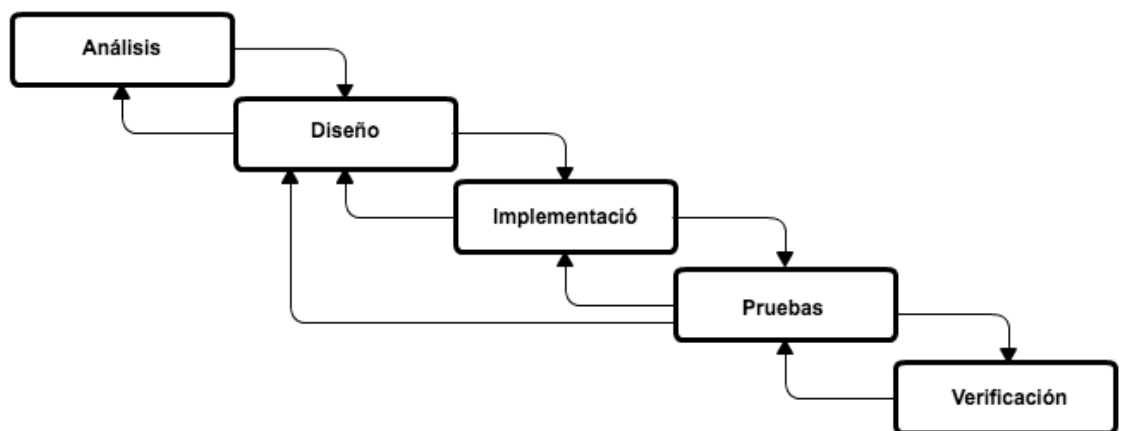


Figura 17. Fases que componen el modelo de desarrollo en cascada.

### 3.2.3. Modelo iterativo-incremental

El modelo de desarrollo iterativo-incremental está basado en el modelo anteriormente descrito, de desarrollo en cascada. Este proceso de desarrollo software [31] permite cubrir en gran medida las debilidades del modelo de desarrollo en cascada, como puede ser la rigidez secuencial de las etapas o el tiempo para corrección de errores en etapas avanzadas del proyecto, entre otras.

Un proyecto que siga este modelo, es planificado en distintas etapas temporales denominadas iteraciones, en las cuales se repite un determinado proceso de trabajo que comienza con el análisis y finaliza con las pruebas. Cada iteración da como resultado una versión más completa y cercana al producto final. Este modelo, Figura 18, permite poder trabajar de manera conjunta en las fases de especificación, desarrollo y validación, y que cada versión permita retroalimentar a la siguiente.

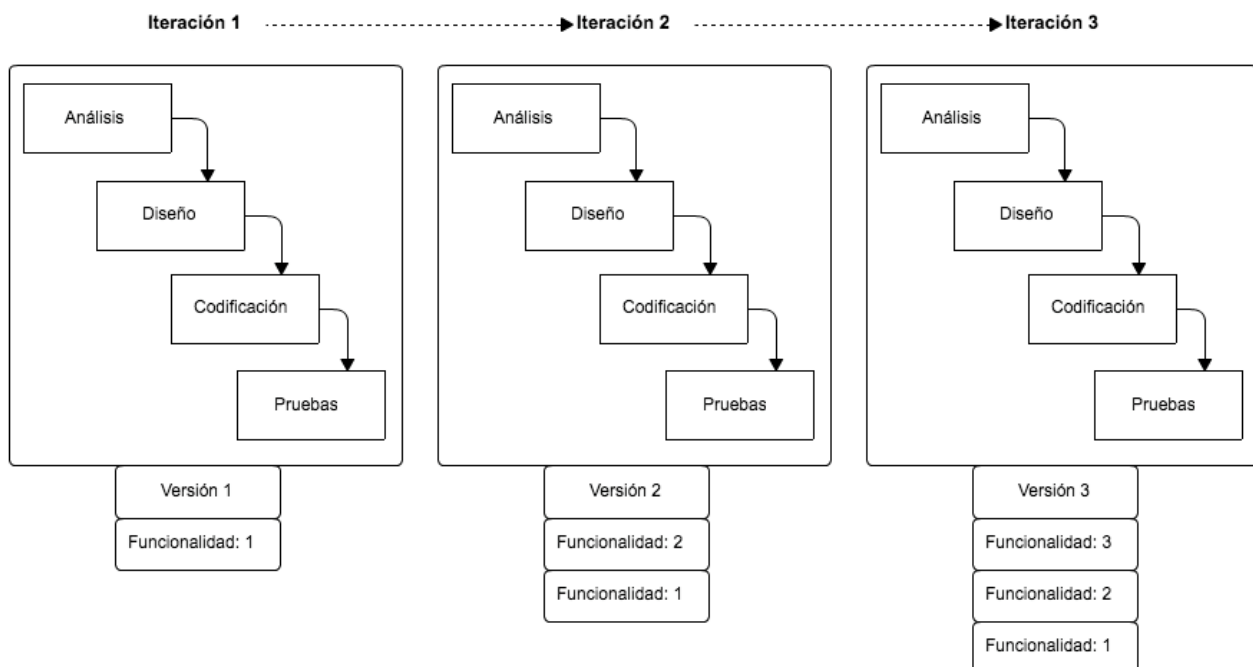


Figura 18. Ciclo de vida iterativo e incremental, en donde cada fase (iteración) sirve de base para la siguiente.

La base del proceso es comenzar con implementaciones simples de los requisitos del sistema y en cada iteración mejorar la versión del proyecto hasta que el sistema completo este implementado. En cada iteración se procede a realizar cambios que permitan agregar nuevas funcionalidades y capacidades al sistema.

### 3.3. Herramientas empleadas para el desarrollo del proyecto

Para el desarrollo de este proyecto se han empleado las siguientes tecnologías:

#### 3.3.1. Entornos virtuales

Un entorno virtual es una herramienta que permite mantener las dependencias necesarias para diferentes proyectos en lugares separados, dentro de un mismo sistema.

**Virtualenv**<sup>27</sup> es una herramienta que permite crear entornos virtuales Python aislados, ya sea del propio sistema principal o de otros entornos virtuales. De esta manera es posible tener instaladas múltiples versiones de una misma librería sin tener ningún tipo de conflicto, Figura 19. Para ello, esta herramienta crea un directorio que contiene todos los ejecutables (librerías y entorno de ejecución) necesarios para usar los paquetes que un proyecto Python pueda necesitar.

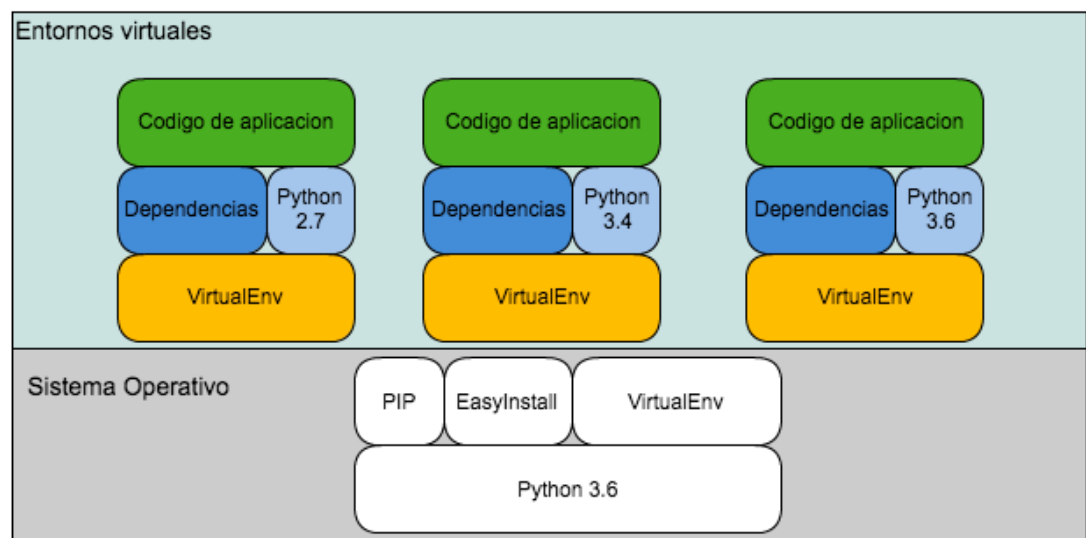


Figura 19. Ejemplo de stack usado en Virtualenv.

#### 3.3.2. Pycharm

Como entorno de desarrollo integrado (IDE) se va a usar *Pycharm*. Se trata de un IDE multiplataforma<sup>28</sup> enfocado al desarrollo en Python. Permite análisis de código, depuración, gestión de intérpretes de Python, soporte para Django, etc. Se va a emplear la versión *community* que es gratuita y enfocada a educación. La Figura 20, es una captura de la interfaz de Pycharm, desde la cual se tiene acceso a todos los recursos necesarios para el desarrollo

<sup>27</sup> VirtualEnv, disponible en: <https://python-guide-pt-br.readthedocs.io/en/latest/dev/virtualenvs/>

<sup>28</sup> Pycharm, disponible en: <https://www.jetbrains.com/pycharm/>

y ejecución de un proyecto. También es capaz de completar el código, y llevar a cabo la corrección y marcado de errores al vuelo

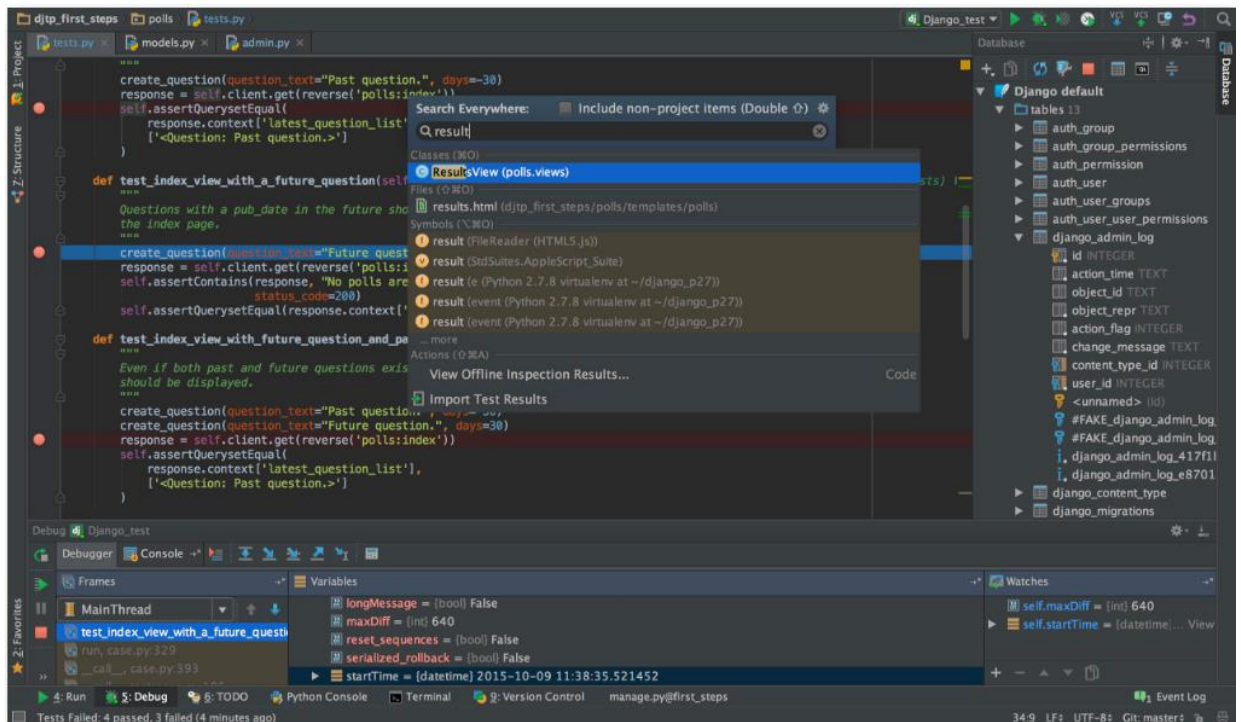


Figura 20. Ejemplo de cómo Pycharm es capaz de completar los comandos

### 3.3.3. Github

Github<sup>29</sup> es una plataforma de desarrollo colaborativo que permite alojar proyectos utilizando el sistema de control de versiones Git [32]. En este proyecto se utiliza la versión gratuita, que permite almacenar código de manera pública. El uso de Git permite: auditar el código, llevar un control de versiones, volver a versiones anteriores de una manera sencilla, aporta seguridad, ya que los ficheros se firman internamente con SHA1, para registrar cualquier cambio. También permite trabajar con ramas para la prueba de distintas funcionalidades (branch), y si son adecuadas volverlas a unir (merge) al código principal (master).

<sup>29</sup> Github, disponible en: <https://github.com/>

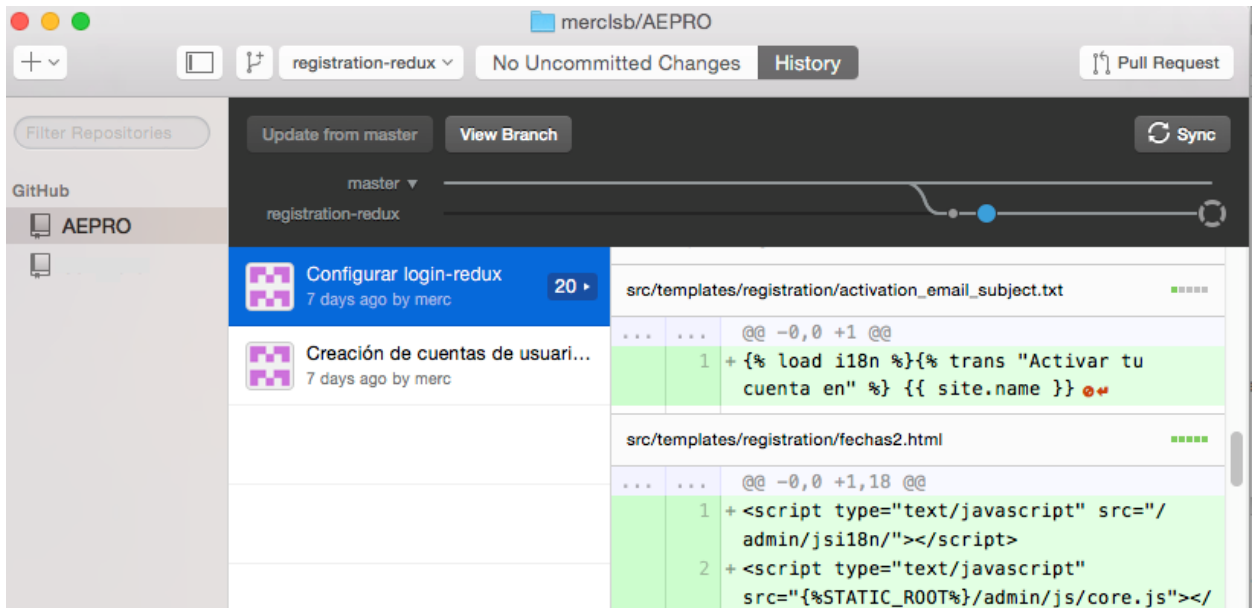


Figura 21. Ejemplo de la aplicación para escritorio Github

### 3.4. Planificación

Una vez definido el alcance del proyecto, y habiendo realizado un estudio de los requisitos, se puede llevar a cabo una planificación de ejecución del proyecto más realista. Aun habiendo realizado esta planificación, durante la fase inicial del proyecto, se han incluido todas las tareas necesarias para alcanzar con éxito los objetivos fijados. En la Figura 22, se puede ver la planificación realizada entre los meses marzo y abril de 2017, para la gestión del proyecto.

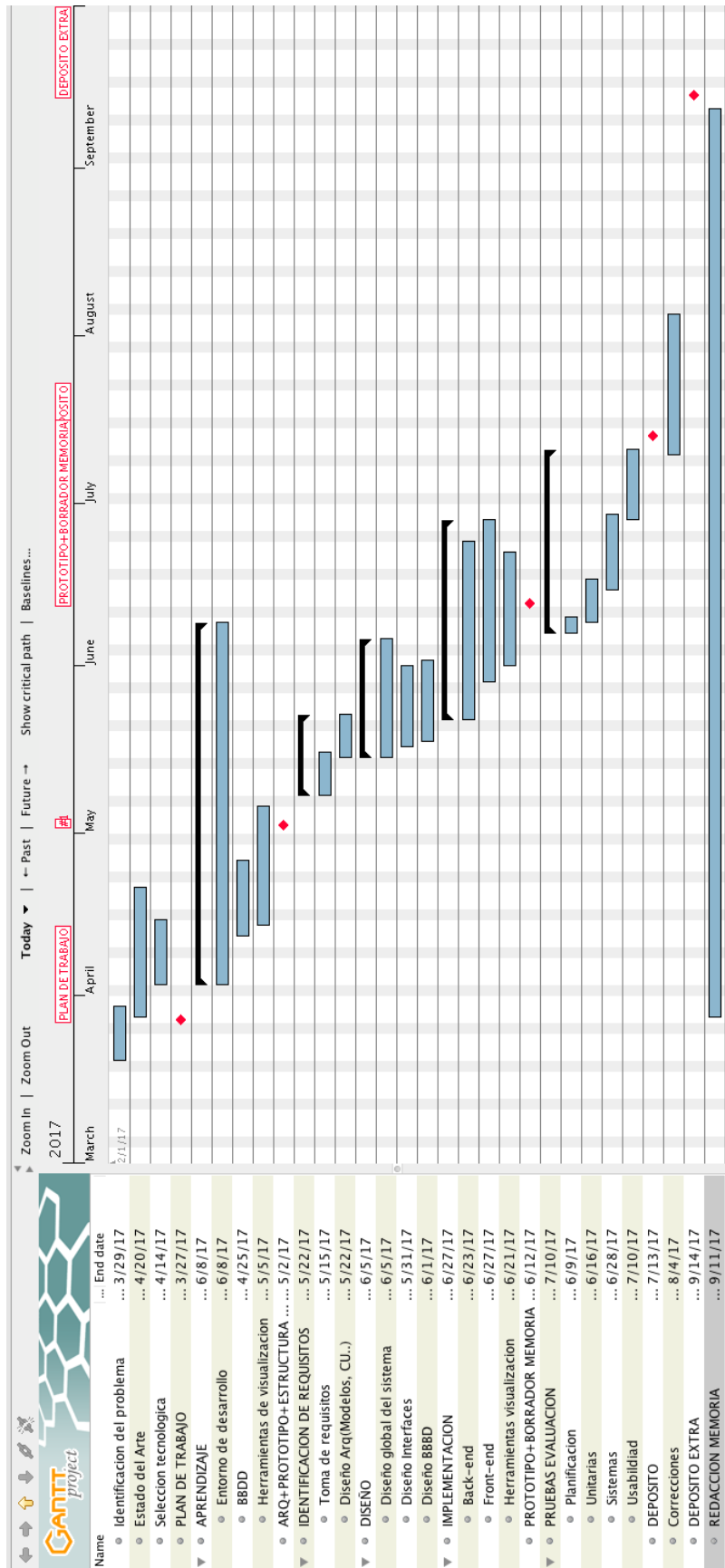


Figura 22. Diagrama de Gantt para la planificación de proyecto

## 4. Diseño de la solución

Realizado el análisis de requisitos que debe cumplir la aplicación, es necesario llevar a cabo un planteamiento de cómo debe estar estructurado el sistema, y diseñar las funcionalidades que debe cumplir. Con el fin de facilitar la comprensión de la estructura que presentará este desarrollo, se ha realizado:

- Un **diseño global del sistema** que permita tener una idea general sobre los flujos de procesos.
- Se ha empleado un **modelo de datos** que permita estructurar la información que el sistema tiene que almacenar.
- Mediante **diagramas de actividad** se muestra las distintas fases que siguen los procesos del sistema.
- Un **mapa de navegación** que permita estructurar la navegación por la aplicación.
- El uso de **prototipos de la interfaz** de usuario que permitan dar una idea lo más cercana posible del diseño final de la aplicación.

### 4.1. Diseño global del sistema

Mediante el siguiente diagrama, Figura 23, se pretende dar una aproximación general del sistema a implementar. En ésta figura se pueden diferenciar la parte correspondiente al *front-end* en el cual, mediante distintos módulos, el usuario puede interactuar con el sistema:

- Módulo de **creación de nuevos análisis**, contemplado en el RQF-1.
- Módulo de **comprobación de formato de archivos**, contemplado en el RQF-6.
- Módulo de **información de usuario**, contemplado en el RQF-21.
- Módulo de **lista de resultados**, contemplado en el RQF-8.

La otra parte del diagrama, el *back-end*, es el encargado de procesar, almacenar y recuperar los datos. Los dos primeros módulos que permiten la **creación de nuevos análisis** y **comprobación formatos de archivos** tienen en común la necesidad de realizar un procesamiento sobre la información proporcionada por el usuario, descritos en los RQF 2, 3 y 4. Para el módulo de **creación de nuevos análisis**, una vez superado la validación de datos, la información es procesada y los resultados obtenidos almacenados en la base de datos. De esta forma, mediante el módulo de **lista de resultados** el usuario puede visualizar los análisis creados previamente, descrito en los RQF 8, 9, 10 y 11, y acceder al contenido de los resultados, que le serán mostrados mediante distintos gráficos, considerados en los RQF 12, 13, 14, 15, 16.

Por último, el módulo **información de usuario**, guarda información relativa al usuario en la base de datos y le permite modificarla en caso de ser necesario, contemplado en los RQF 19, 20, 21.

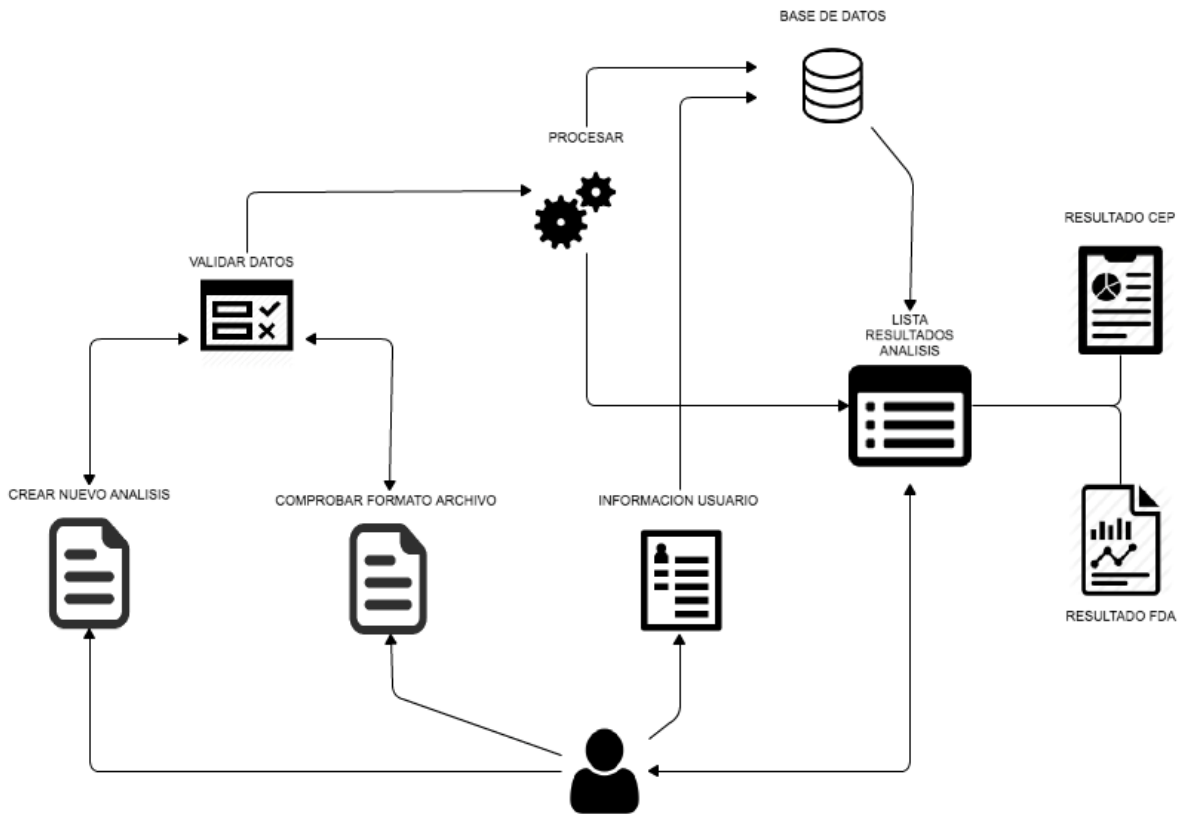


Figura 23. Aproximación del diseño global del sistema

## 4.2. Modelo de datos

También es necesario definir las necesidades de almacenamiento de los datos. El uso de modelado de datos, permite organizar los datos en el sistema, permitiendo crear de una manera rápida y sencilla una base de datos que cumpla con las necesidades de almacenamiento. La representación de los datos, que estarán en el sistema, se realiza mediante de un diagrama Entidad-Relación, Figura 24, que permite mostrar la definición de entidad y las relaciones entre los datos.

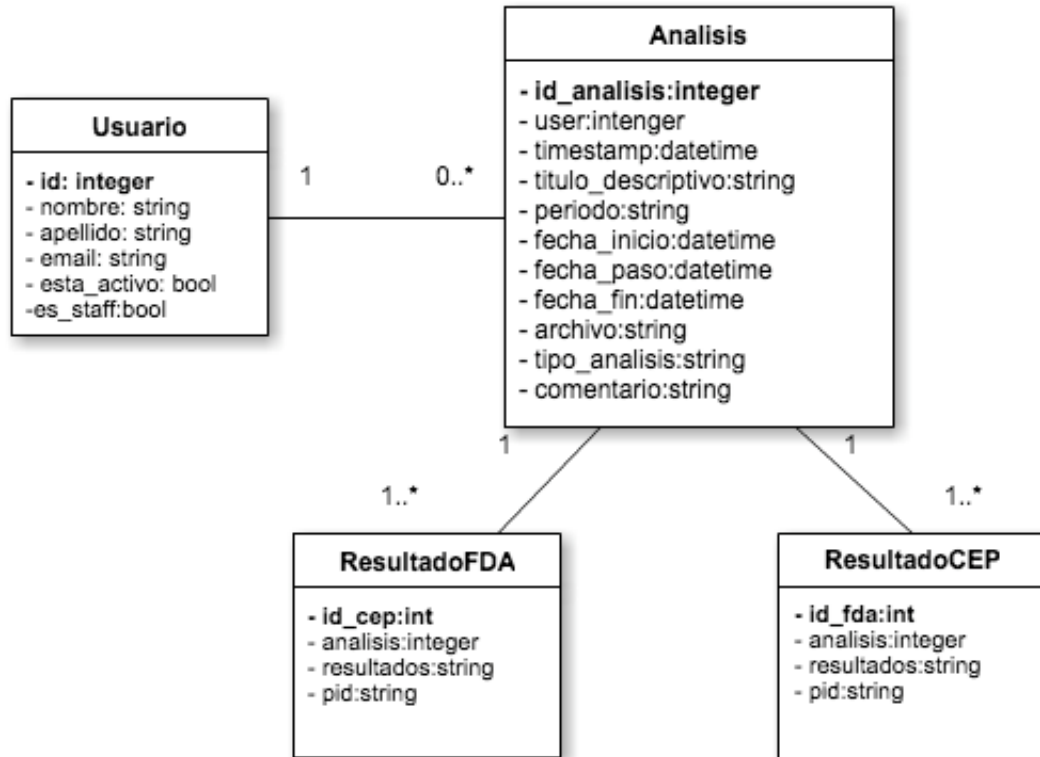


Figura 24. Diagrama Entidad-Relación AEPRO

A continuación, se procede a describir los componentes del diagrama Entidad-Relación, sus entidades, atributos y relaciones, tal y como se representan en la Figura 24.

<b>Entidad</b>	<b>Usuario</b>
<b>Representación</b>	Representa a cada uno de los usuarios del sistema.
<b>Atributos</b>	<ul style="list-style-type: none"> <li>- id: identificador único asociado a cada usuario.</li> <li>- nombre: nombre del usuario</li> <li>- apellido: apellido del usuario</li> <li>- email: email del usuario</li> <li>- esta_activo: variable para identificar si una cuenta esta activa o no.</li> <li>- es_staff: variable para indicar si el usuario pertenece al grupo staff.</li> </ul>

Tabla 36. Entidad Usuario y sus atributos.

<b>Entidad</b>	<b>Análisis</b>
<b>Representación</b>	Representa cada una de las instancias que serán generadas por los usuarios.
<b>Atributos</b>	<ul style="list-style-type: none"> <li>- id_analisis: identificador de cada análisis</li> <li>- user: usuario al que se asocia el análisis</li> <li>- timestamp: marca de tiempo en que se genera el nuevo análisis</li> <li>- titulo_descriptivo: descripción realizada por el usuario</li> <li>- periodo: sobre el que realizar el análisis los valores son: día, mes, o año</li> <li>- fecha_inicio: fecha de inicio de la medición</li> </ul>

	<ul style="list-style-type: none"> <li>- fecha_paso: intervalo entre la primera y segunda medición.</li> <li>- fecha_fin: fecha de final de medición</li> <li>- archivo: fichero proporcionada por el usuario</li> <li>- tipo_analisis: los valores son CEP o/y FDA</li> <li>- comentario: anotación realizada por el usuario</li> </ul>
--	--

Tabla 37. Entidad Análisis y sus atributos

<b>Entidad</b>	<b>ResultadoFDA</b>
<b>Representación</b>	Representa toda la información asociada al análisis, generada a partir de los resultados obtenidos del proceso de análisis funcional de datos.
<b>Atributos</b>	<ul style="list-style-type: none"> <li>- id_fda: identificador del resultado</li> <li>- analisis: identificador del análisis al que se asocia el resultado</li> <li>- resultados: variable en la que almacenar toda la información para la representación.</li> <li>- pid: identificador del proceso y estado</li> </ul>

Tabla 38. Entidad ResultadoFDA y sus atributos

<b>Entidad</b>	<b>ResultadoCEP</b>
<b>Representación</b>	Representa toda la información asociada al análisis, generada a partir de los resultados obtenidos del control estadístico de procesos.
<b>Atributos</b>	<ul style="list-style-type: none"> <li>- id_cep: identificador del resultado</li> <li>- analisis: identificador del análisis al que se asocia el resultado</li> <li>- resultados: variable en la que almacenar toda la información para la representación.</li> <li>- pid: identificador del proceso y estado.</li> </ul>

Tabla 39. Entidad ResultadoCEP y sus atributos

### 4.3. Diagrama de actividad

Mediante los siguientes diagramas (Figura 25, Figura 26, Figura 27, Figura 28) se trata de representar de una manera más funcional la secuencia de actividades que se llevan a cabo en el sistema y las condiciones que generen distintos estados del sistema.

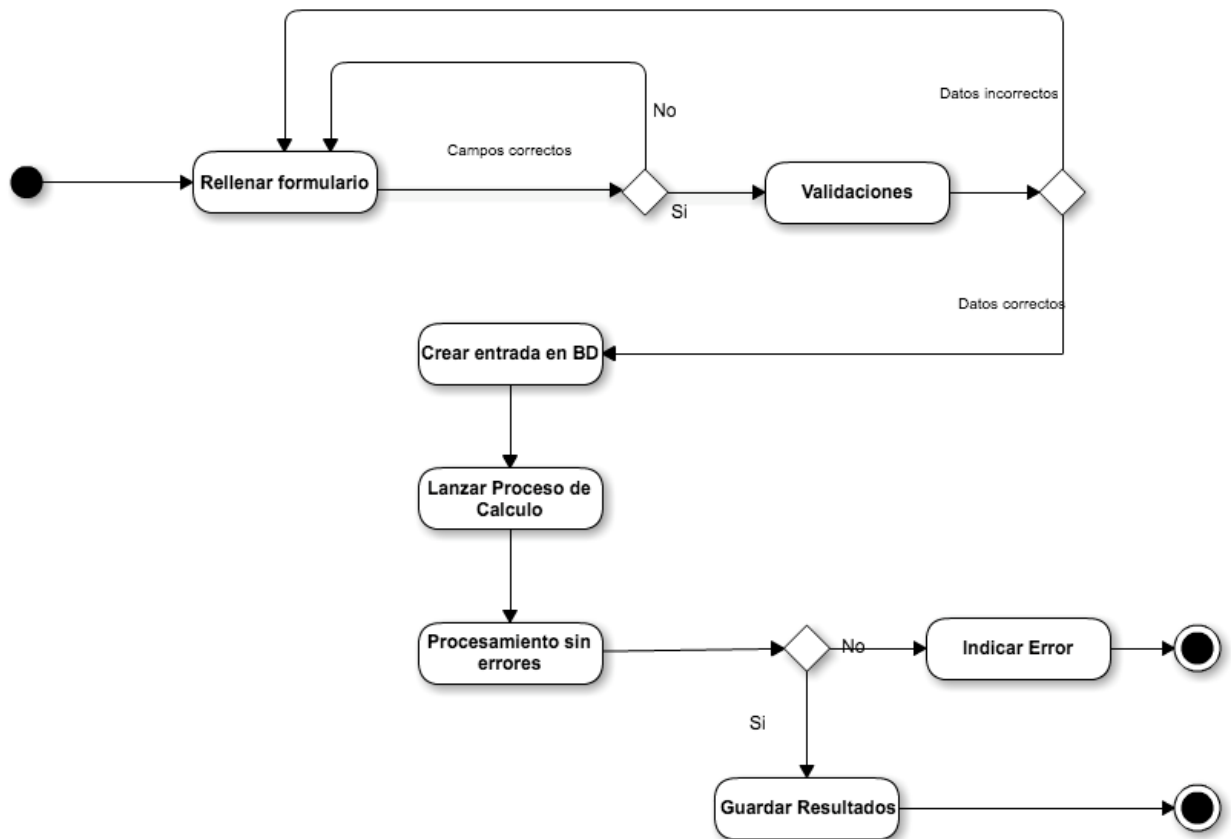
**Diagrama de actividad: Proceso de creación de un nuevo análisis**

Figura 25. Diagrama de actividad: Proceso de creación de un nuevo análisis

**Flujo de control**

El proceso comienza **rellenando un formulario** que contendrá la información necesaria para crear un nuevo análisis. Antes de proceder a realizar las validaciones sobre los datos introducidos se **comprueba** que los **campos** cumplan con una serie de requisitos como: que no estén en blanco, que sean tipo concreto (string, numérico, etc).

Una vez comprobado que los datos son correctos, se procede a aplicar cuatro tipos de **validaciones sobre ciertos campos**, como son las fechas y el fichero, que se detallan en el siguiente diagrama de actividad.

Pasadas estas validaciones se procede a **crear una nueva instancia en la base de datos** y **lanzar un proceso de cálculo**.

Finalizado el procesamiento de los datos se plantean dos posibles finales:

- Que el procesamiento haya terminado de manera satisfactoria, por lo cual los datos resultantes son almacenados en la base de datos.
- Que el procesamiento haya terminado de manera insatisfactoria, por lo cual se indique que se ha producido algún tipo de error.

**Diagrama de actividad: Proceso de validación de un fichero.**

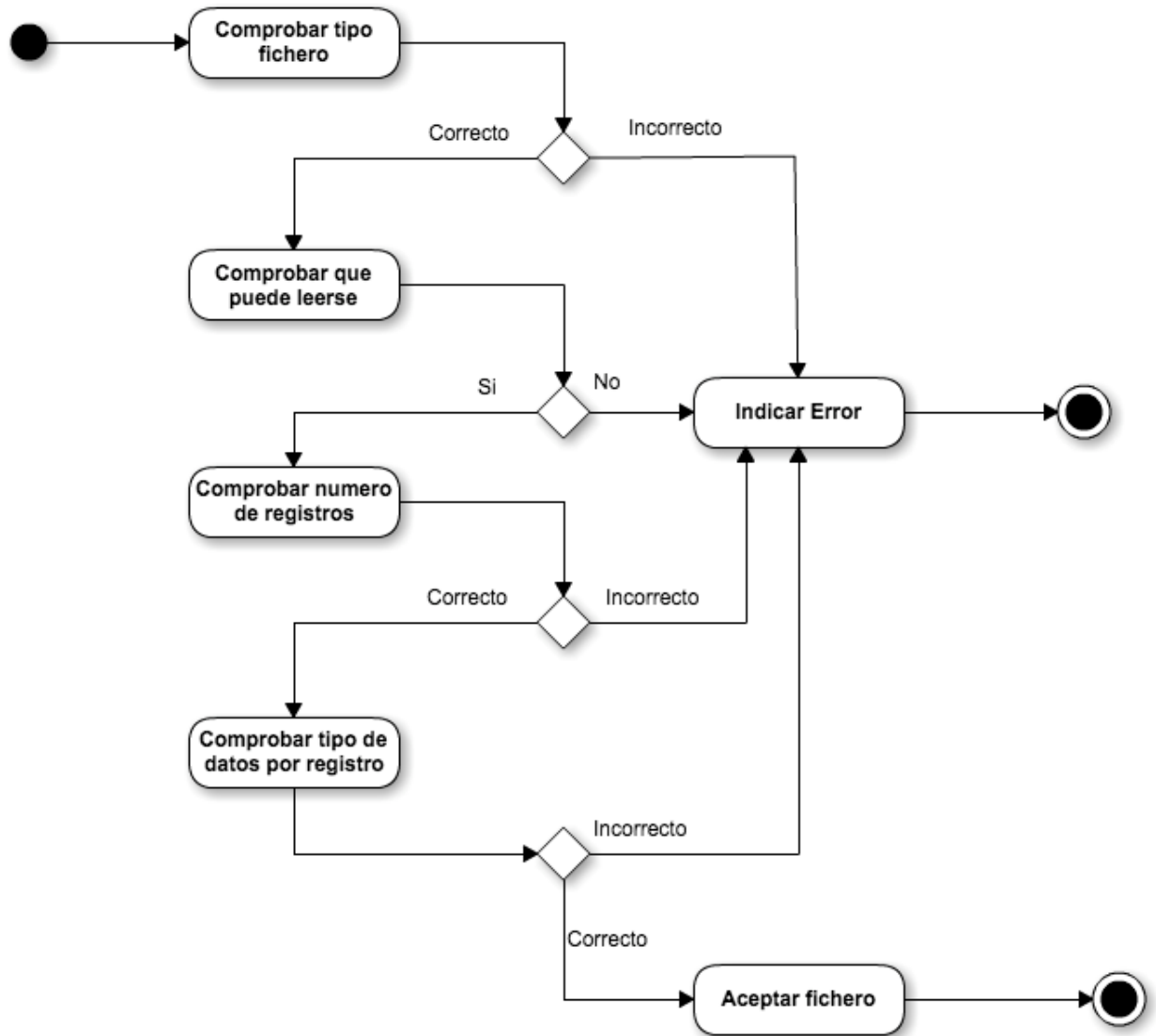


Figura 26. Diagrama de actividad: Proceso de validación de un fichero.

### Flujo de control

El proceso de validación de un fichero consta de las siguientes fases:

- Se comprueba que el fichero es de un tipo en concreto predefinido: xls, en este caso, pero pudiendo ser otros como csv, txt, etc.
- En el caso de no ser uno de los predefinidos, se indica el error correspondiente.
- Si el formato es el correcto, se intenta acceder en modo lectura al contenido.
- En caso de no poder accederse al documento, se indica el error correspondiente
- Una vez que se puede acceder al contenido del archivo se identifican el número de registros con los que cuenta, si no llegan a un mínimo no es posible realizar los cálculos necesarios.
- En el caso de no tener un número mínimo de registros, se indica el error correspondiente.
- Comprobado el número de registros, se procede a comprobar el formato de cada registro: fecha y numérico o vacío, respectivamente.
- Si se encuentra algún otro tipo de dato, se indica el error correspondiente.

### Diagrama de actividad: Proceso de visualización de análisis realizados y resultados obtenidos

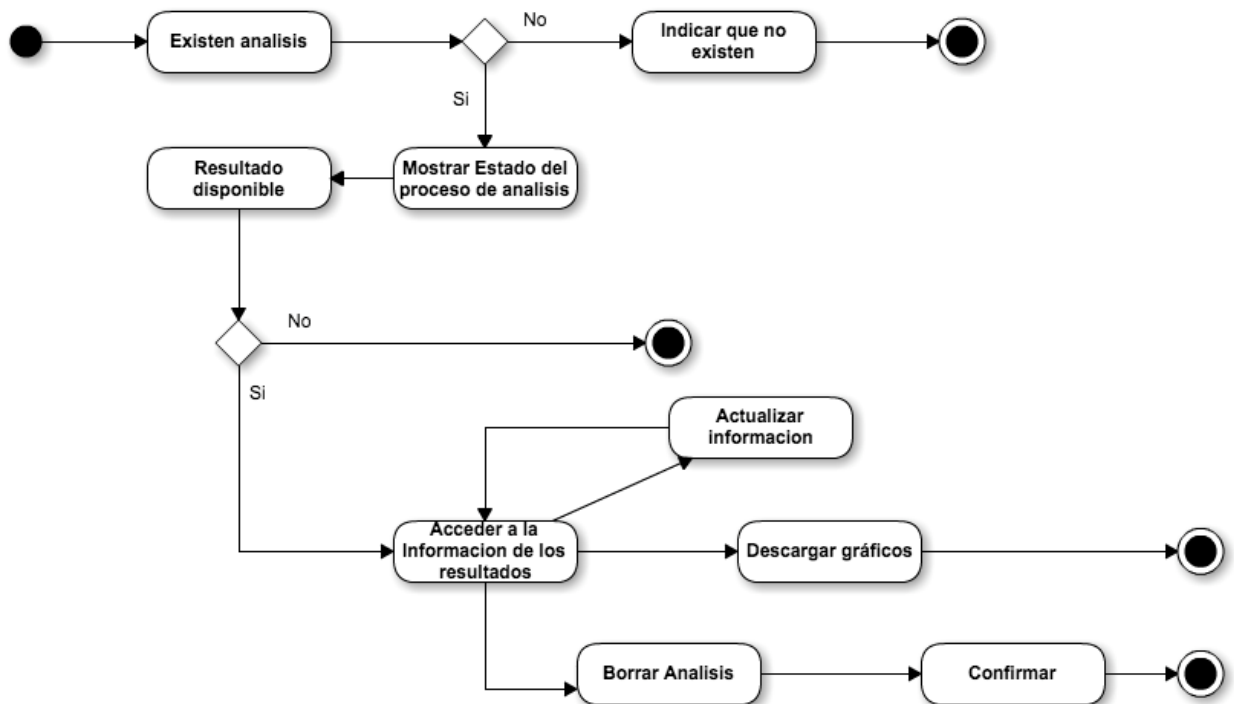


Figura 27. Diagrama de actividad: Proceso de visualización de análisis realizados y los resultados obtenidos

### Flujo de control

Mediante el diagrama de la Figura 27, se pretende desarrollar el proceso que permite mostrar el listado de análisis generados, así como su estado.

- Si al acceder al listado ya existe algún análisis creado anteriormente se mostrará la información correspondiente a cada uno de ellos, así como su estado.
- En el caso de que no haya ningún todavía, se indicará que no existen resultados disponibles.
- Si el proceso de cálculo ya ha finalizado y los resultados se encuentran disponibles será posible acceder a la información.
- Si no ha finalizado, porque se está procesando todavía o a ocurrido algún tipo de error, se indicará.
- Una vez que se pueda acceder a los resultados, se podrá actualizar la información, descargar los gráficos o borrar el análisis y los resultados asociados.

### Diagrama de actividad: Proceso de alta y gestión de usuarios.

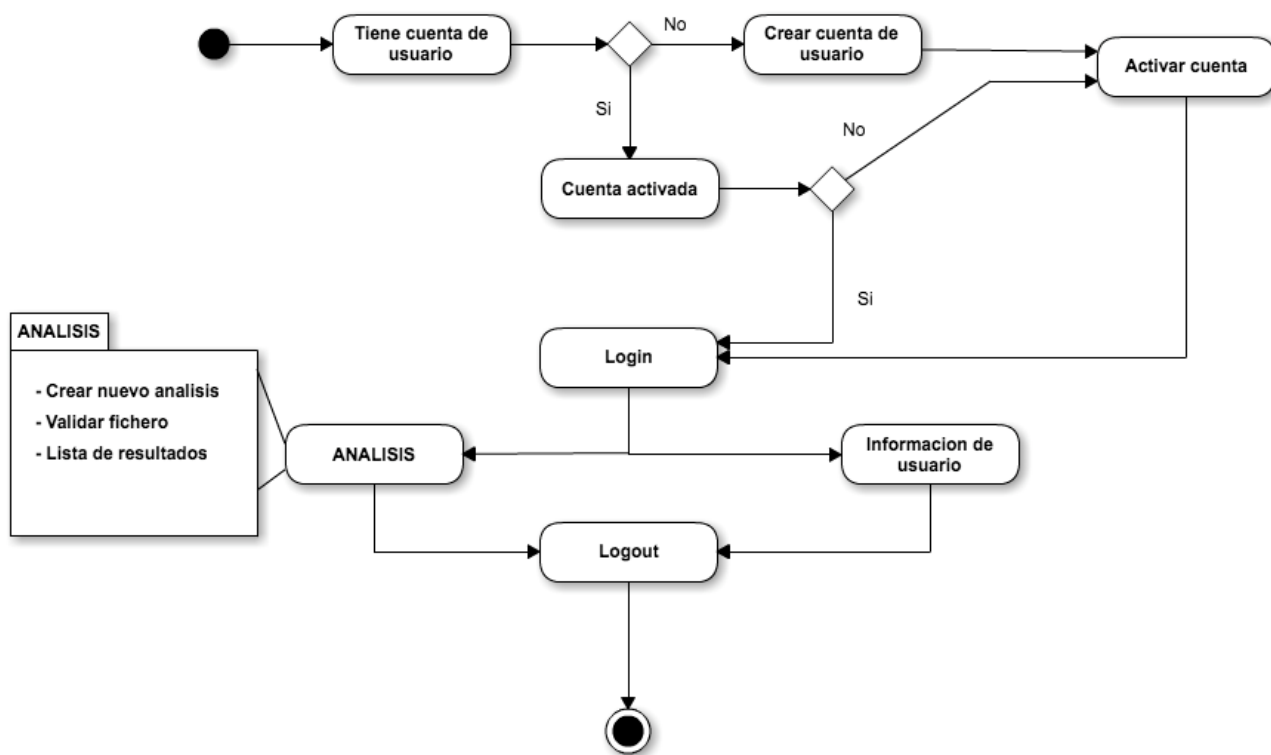


Figura 28. Diagrama de actividad: Proceso de alta y gestión de usuarios.

### Flujo de control

El diagrama representado en la Figura 28, describe el flujo de actividades que sigue la creación y autenticación de un usuario en el sistema.

- Cuando un usuario accede a la aplicación tiene la posibilidad de ingresar con sus credenciales si ya tiene una cuenta de usuario y esta activada, o por el contrario crearse una nueva cuenta y activarla.
- Realizado el proceso de autenticación, el usuario podrá acceder a toda la información que esté asociada a él. Pudiendo ser la información de su perfil o las funcionalidades anteriormente descritas y que forman parte del bloque de análisis.

#### 4.4. Mapa de navegación

Para la representación del mapa de navegación, se ha planteado una navegación jerárquica o de árbol. Como se puede ver en la Figura 29, la estructura comienza con una página principal, y presentan varias opciones que permiten visualizar contenidos más específicos. El desarrollo del contenido se presenta en forma ramifica lo que permite visitar cada una de las secciones por separado. Para el desarrollo tanto del mapa de navegación como de los prototipos que se presentan en el siguiente apartado, Diseño de la interfaz de usuario, se ha empleado la herramienta *Balsamiq Mockus 3*<sup>30</sup>.

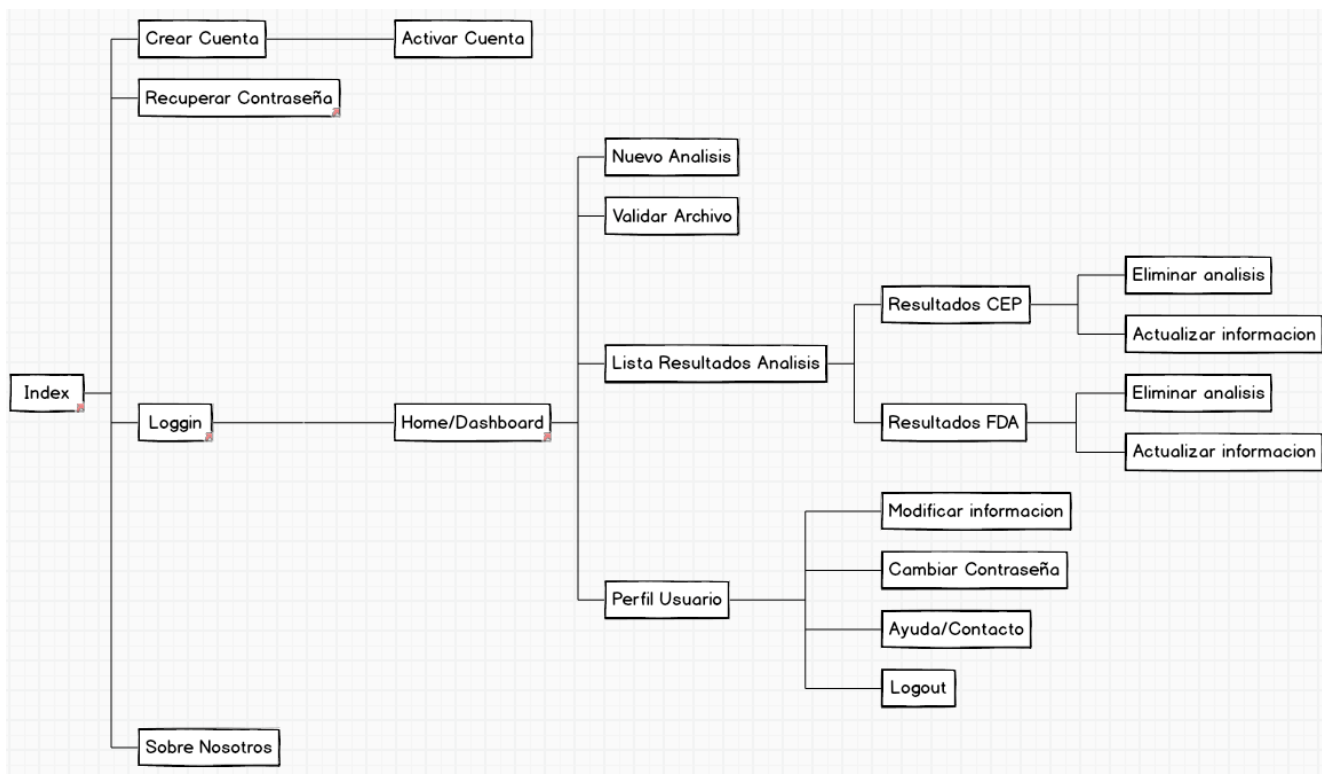


Figura 29. Mapa navegación del sitio web,

<sup>30</sup> Balsamic Mockups 3, disponible en: <https://balsamiq.com/>

Como puede verse en el mapa de navegación, éste presenta un diseño que permite diferenciar rápidamente la estructura que presentará la aplicación. Se pueden identificar dos zonas: una externa y otra interna. La zona externa relativa al *index*, alta de usuarios y *login*, y la zona interna correspondiente a la aplicación de análisis y el perfil del usuario.

#### 4.5. Diseño de la interfaz de usuario

Atendiendo a todas las necesidades analizadas hasta el momento: requisitos, tanto funcionales como no funcionales, casos de uso, diseño global del sistema, diagramas de actividad, y mapa de navegación, se van a desarrollar los siguientes prototipos de interfaz, que servirán como guía para el proceso de implementación. Por tanto, se plantea un diseño que permita mantener una estructura coherente de la interfaz. Ciertos elementos descritos en la Figura 30, como: el menú de navegación, los encabezados, contenidos y los pies de página, son elementos comunes en la mayoría de las aplicaciones web.

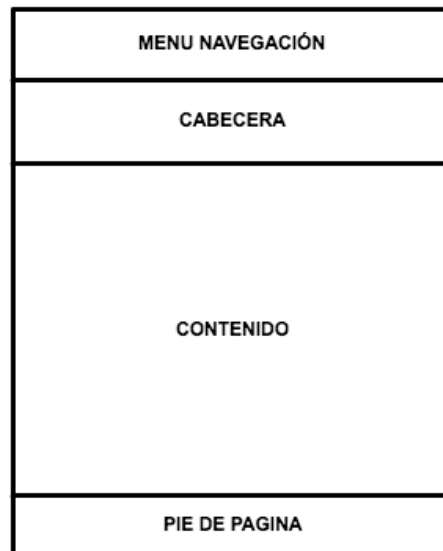


Figura 30. Estructura de la interfaz de usuario.

Atendiendo a la estructura planteada en este apartado se van a presentar varios prototipos, considerados como más destacables para el diseño de la aplicación. El resto pueden verse en el

Anexo II. Prototipos de la interfaz de **usuario**.

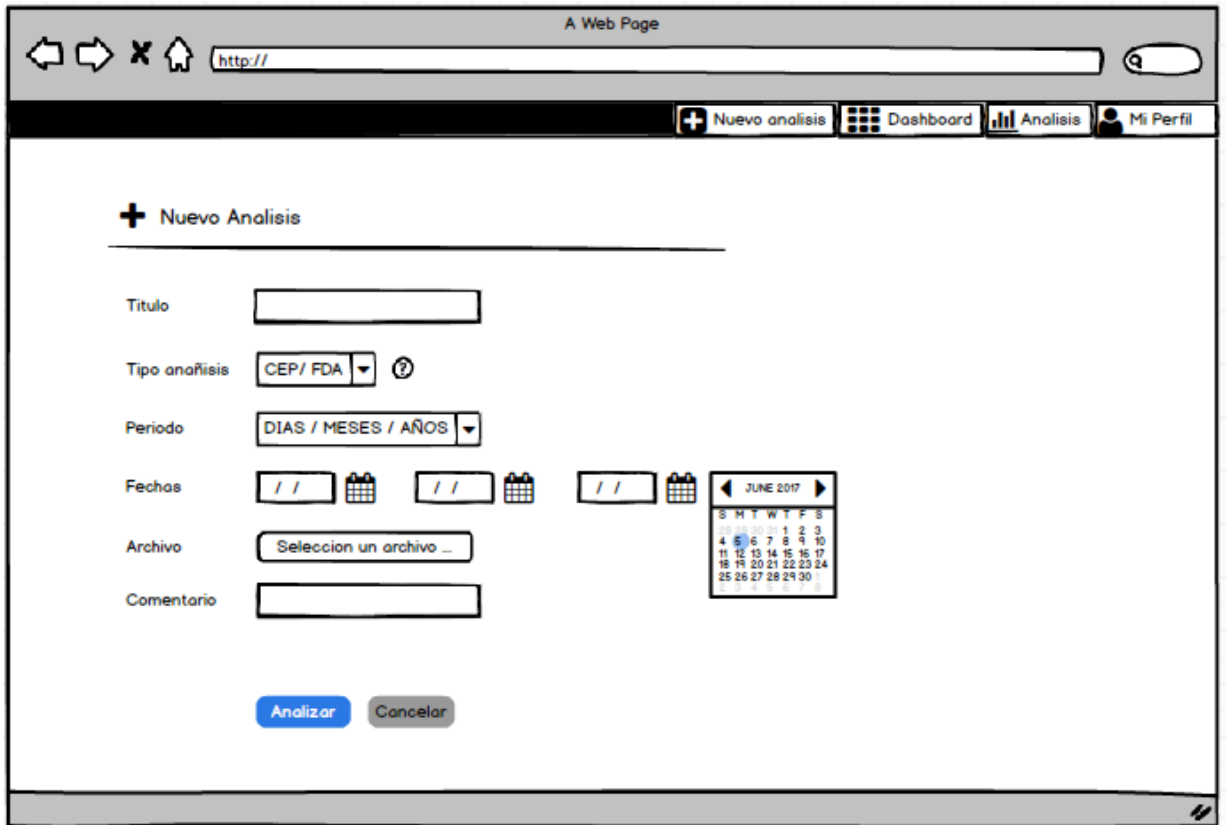


Figura 31. Mockup de la página para crear nuevo análisis

<b>Pantalla</b>	<b>Crear nuevo análisis</b>
<b>Descripción</b>	Pantalla desde la cual podrá crear un nuevo análisis. Para ello es necesario completar toda la información solicitada.
<b>Requisitos funcionales asociados</b>	RQF - 1, RQF - 2, RQF - 3, RQF - 4, RQF - 5, RQF - 6
<b>Requisitos No funcionales asociados</b>	RQNF - 1, RQNF - 4, RQNF - 5, RQNF - 6

Tabla 40. Descripción y requisitos asociados a la interfaz de creación de un nuevo análisis.

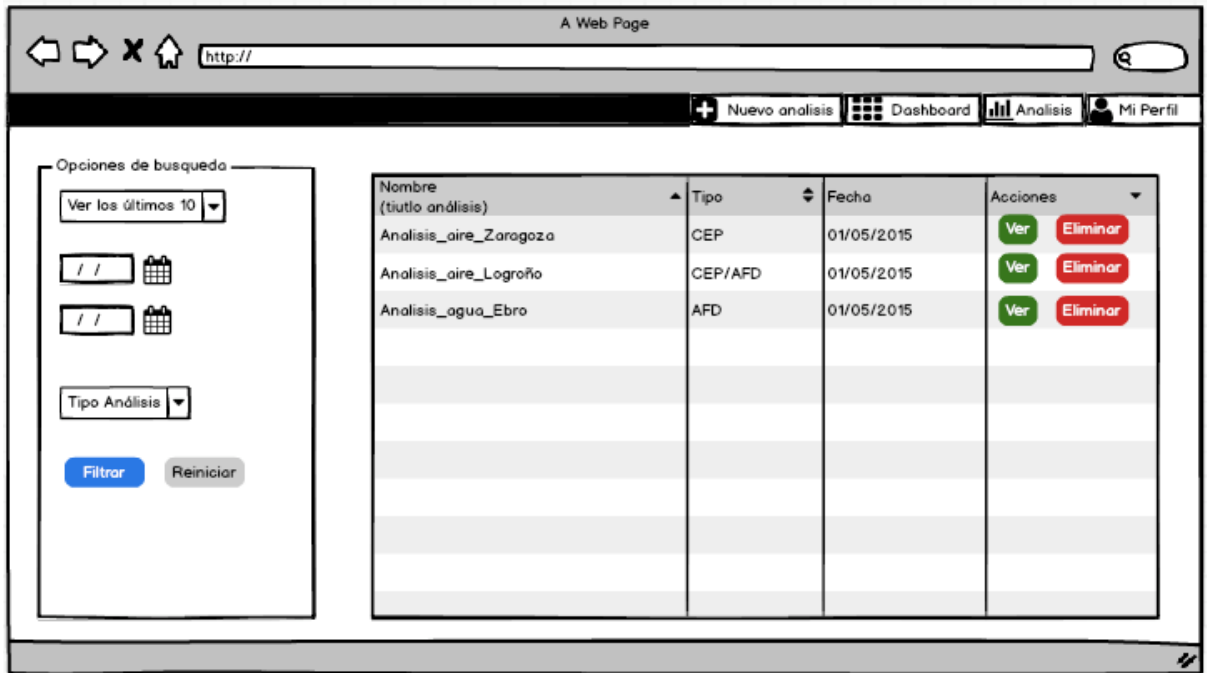


Figura 32. Mockup de la página que muestra listado de análisis creados.

<b>Pantalla</b>	<b>Listado de los análisis creados</b>
<b>Descripción</b>	Pantalla muestra la información de aquellos análisis que el usuario ha ido creando. Se puede el nombre, el tipo de análisis, la fecha de creación. Las acciones a realizar son ver la información del análisis o eliminarlo
<b>Requisitos funcionales asociados</b>	<b>RQF - 7, RQF - 8, RQF - 9, RQF - 10, RQF - 11, RQF - 12</b>
<b>Requisitos No funcionales asociados</b>	<b>RQNF - 2</b>

Tabla 41. Descripción y requisitos asociados al listado de análisis generados.

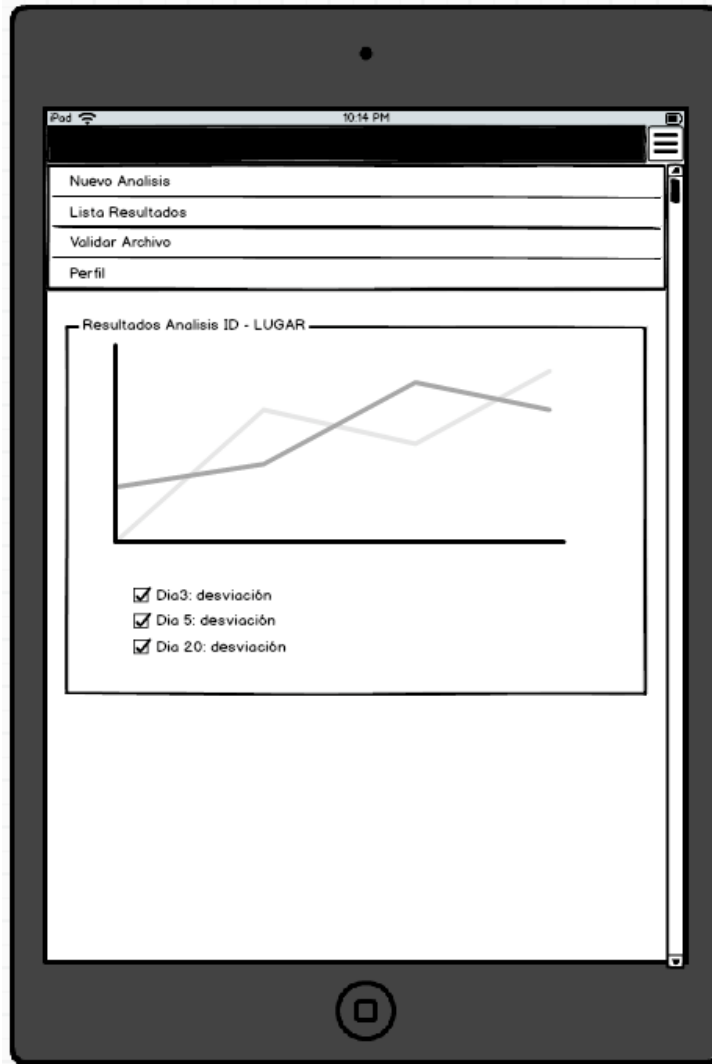


Figura 33. Mockup Resultados del análisis funcional de datos en un dispositivo móvil

Pantalla	Resultado de un Análisis Funcional (FDA)
Descripción	Pantalla muestra la información de aquellos análisis que el usuario ha ido creando. Se puede el nombre, el tipo de análisis, la fecha de creación. Las acciones a realizar son ver la información del análisis o eliminarlo. También se puede comprobar observar que la aplicación es capaz de adaptarse a un dispositivo móvil
Requisitos funcionales asociados	RQF - 15, RQF - 16
Requisitos No funcionales asociados	RQNF - 3

Tabla 42. Descripción y requisitos asociados a la interfaz de resultados de un análisis FDA visto en un dispositivo móvil.

## 5. Descripción técnica de la solución

En este capítulo se describe como se ha llevado a cabo la implementación de la aplicación, mostrando los aspectos más importantes de la fase de desarrollo, una vez realizado el análisis y diseño del sistema, presentado las tecnologías empleadas para la implementación del proyecto

### 5.1. Descripción funcional de la aplicación

Esta aplicación web tendrá que ser capaz de validar un fichero de datos, realizar un análisis estadístico de dichos datos, generando unos resultados que serán representados mediante gráficos estadísticos interactivos que ayuden al usuario (experto) en la interpretación de los mismos y en la toma de decisiones.

Por tanto, la aplicación tiene que permitir que los usuarios puedan cargar un fichero correspondiente a una serie temporal, cuyos datos han sido extraídos generalmente de dispositivos de medida automáticos de indicadores de calidad del agua, o/y del aire.

Estos registros son volcados en un fichero cuya estructura básica es:

- Una columna de tipo fecha, donde se registra el instante de en cual se ha tomado la muestra o se ha realizado la medición.
- Una columna asociada al instante, con una variable de tipo numérica, con el valor de la muestra adquirida. (O<sub>2</sub>, N<sub>2</sub>, ...)

A su vez se podrá seleccionar el periodo temporal (días, meses, años) sobre el que realizar alguno de los tipos de análisis que se describen a continuación.

Para poder llevar a cabo el análisis de los datos proporcionados por los usuarios, la aplicación permitirá realizar dos tipos de análisis estadísticos: un control estadístico de procesos (CEP) y un análisis funcional de datos (AFD).

**Análisis CEP:** este tipo de análisis trata los datos a estudiar (días, meses, años) en función de sus valores medios (resumiendo mucho la información), y engloba el análisis descriptivo clásico y el análisis del concepto de control estadístico (variabilidad).

**Análisis descriptivo clásico:** permite conocer las características más importantes de los datos suministrados: media, mediana, moda, varianza... y los *outliers* puntuales (aquellos elementos anómalos frente a la muestra proporcionada). Como resultado de este tipo de análisis serán las siguientes graficas:

- *Boxplot*,
- Histograma,
- Probabilidad normal
- Autocorrelación.

Véase como ejemplo la Figura 34, en la cual se muestra un boxplot en el que se puede obtener la información relativa a los diferentes cuartiles de la muestra, y un histograma del que se puede inferir el tipo de distribución que siguen los datos (normal, weibull, lognormal, valor, etc.)

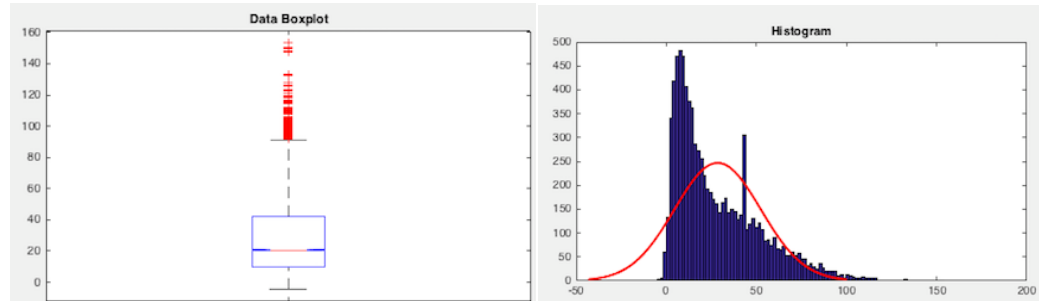


Figura 34. Ejemplo de un Boxplot(izq) y de un histograma (drch).

**Análisis de control de calidad:** mediante este tipo de análisis el usuario podrá determinar si su proceso está bajo control, es decir en ausencia de variabilidad especial y únicamente sometido a variabilidad común, así como la capacidad del mismo para alcanzar los límites óptimos. Se trata de un análisis empleado en procesos industriales, y se basa en el análisis descriptivo de los datos para establecer unos límites que determinen los outliers (evidencias de variabilidad asignable) de un proceso.

Por tanto, del análisis CEP el usuario podrá determinar:

- Si un proceso está bajo control estadístico, es decir si la variabilidad del proceso es únicamente aleatoria, o es provocada por algún factor trazable (especial).
- La capacidad del mismo (pudiendo estar el proceso bajo control, pero no ser capaz de alcanzar unos límites marcados por la tolerancia). Como resultado de este tipo de análisis se generarán los gráficos de control XBAR-S, Figura 35, de donde se puede obtener información relativa a causas de variabilidad especial o asignable.

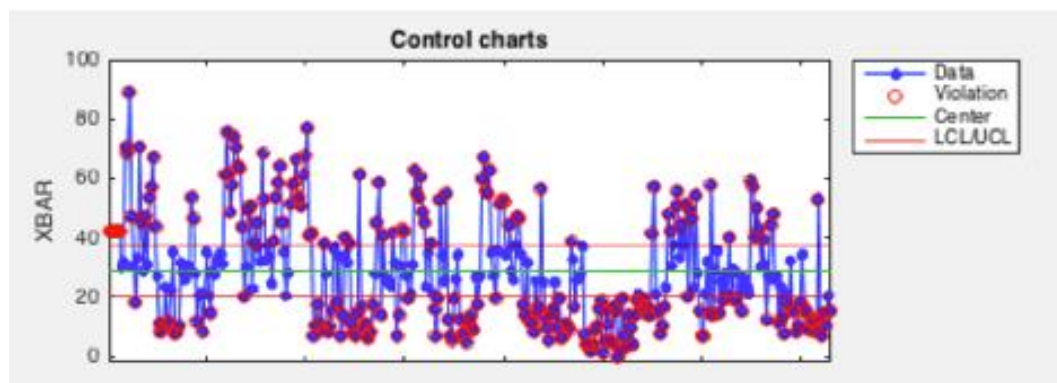


Figura 35. Ejemplo de un gráfico XBAR: superior representación de las medias. Inferior representación de las desviaciones.

**Análisis funcional:** este tipo de análisis permite tener en cuenta toda la información del periodo a analizar, siendo, a su vez, insensible a errores de medida en un porcentaje muy alto de los casos, que pueden ser considerados falsas alarmas en el análisis clásico.

Esto es debido a que este tipo de análisis trata los datos como funciones (curvas continuas) y no como un vector de datos, por lo que no se pierde información, permitiendo analizar la serie de manera continuada en el tiempo.

El resultado de este análisis, muestra los *outliers* (días, meses, años) en los cuales se ha producido algún tipo de desviación. La identificación de estos *outliers* se hace en base a la profundidad funcional, es decir a la desviación de cada una de las curvas de la función media. Como se puede observar en la Figura 36, el análisis funcional permite calcular las curvas representativas de los datos de la concentración de CO, de cada día, para una muestra de un mes (30 días). Como resultado del análisis se representan 30 curvas (gris) y se remarcan (negro) aquellas que indican la presencia de algún comportamiento anómalo en las muestras.

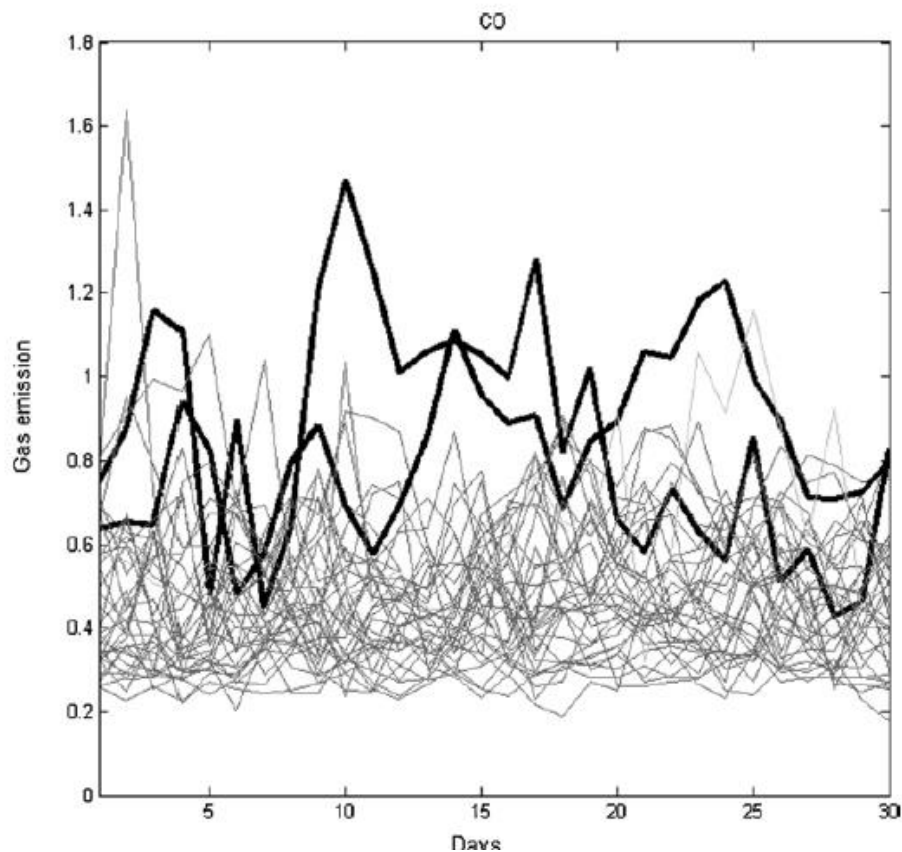


Figura 36. Resultado de un análisis funcional. En negro las curvas que representan los días en los que se ha producido una desviación en las emisiones de gas. [4]

Para el procesamiento de los datos se hace uso de una herramienta de análisis (*Módulo de Análisis Estadístico.- MAE*) [4] implementada en Matlab y basada en la adaptación de un

conjunto de técnicas estadísticas relativas al análisis de la calidad ambiental: Control Estadístico de la Calidad (CEC), Control Estadístico de Proceso (CEP) y Análisis Funcional de datos (AFD) [33].

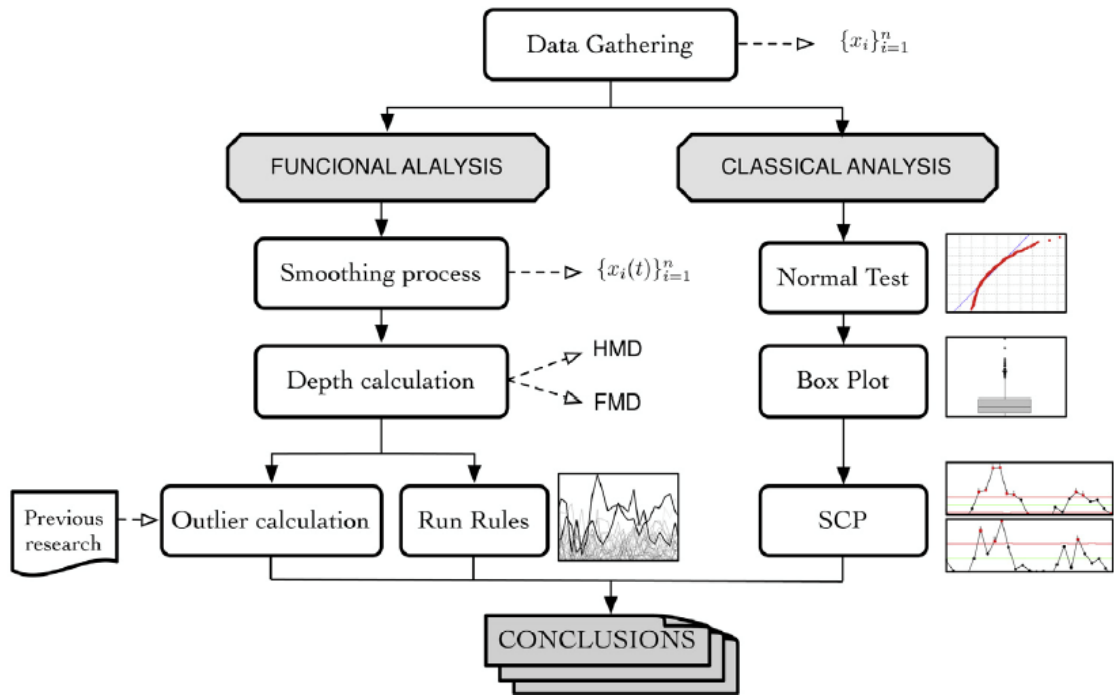


Figura 37. Diagrama general de la metodología, en donde se puede ver las diferentes fases del proceso de análisis. [34]

## 5.2. Tecnologías utilizadas en la implementación

Sobre el diagrama general planteado en la fase de diseño (*Figura 23. Aproximación del diseño global del sistema*) se indican las tecnologías para la implementación de la aplicación, Figura 38.

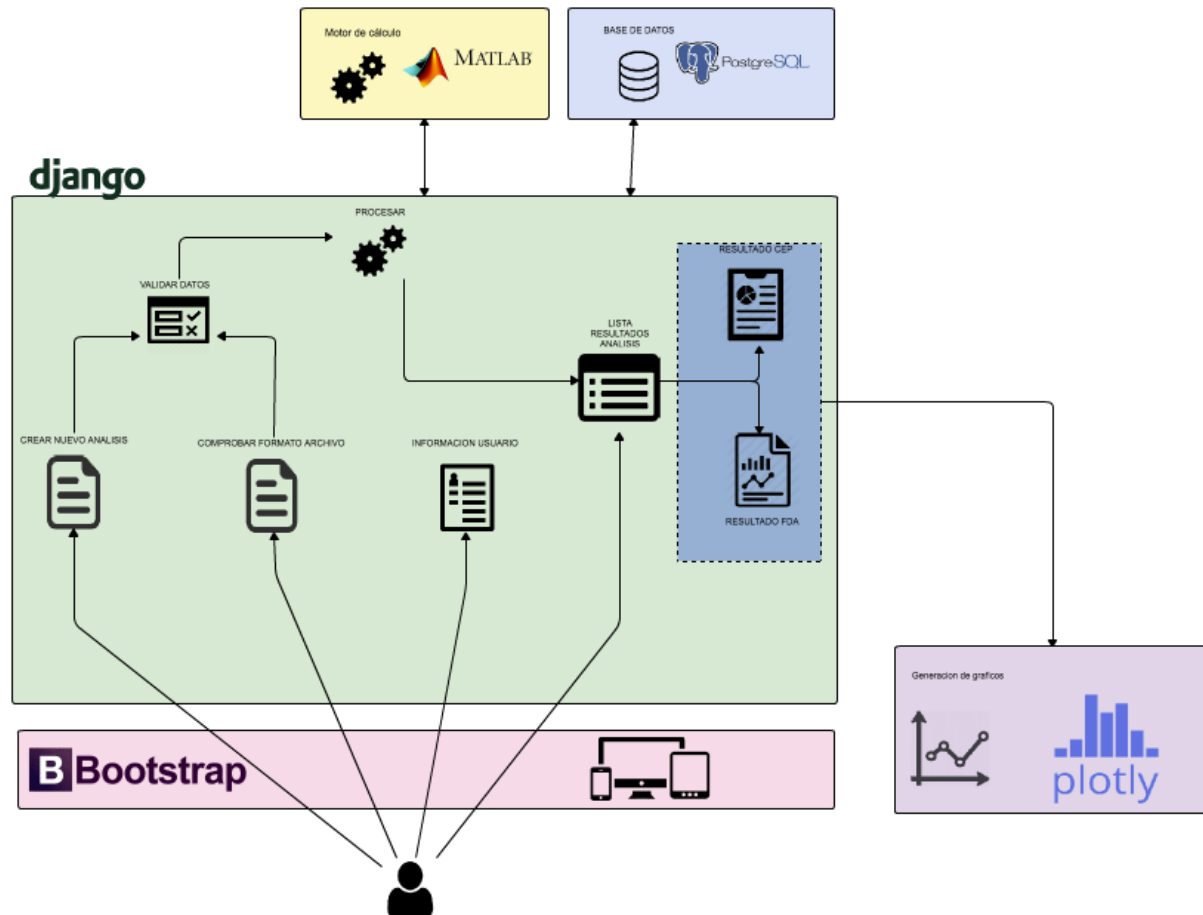


Figura 38. Diagrama general del sistema con las tecnologías aplicadas

### Django, Versión: 1.10

Este framework de desarrollo web en Python representa la parte central de la aplicación. Cuenta con un servidor web ligero ya preparado para realizar la implementación de la aplicación. Además de favorecer el desarrollo rápido, como puede ser mediante el uso de herencia de plantillas, cuenta con su propio ORM, y contiene medidas de seguridad ya implementadas, como la protección contra CSRF. Su implementación nos permite hacer uso de cualquier paquete Python que necesitemos.

**PostgreSQL, Versión: 9.6.1**

En principio, cualquiera de las tres SGBD estudiados (en el apartado 2.3.3.1) hubiera sido aceptable para este desarrollo. Sin embargo, la elección de tipos de datos JSON, para almacenar la información de los resultados de los análisis, decantó la opción hacia PostgreSQL. Django soporta de manera nativa este tipo de dato sobre PostgreSQL y, aunque es posible el uso de tipos de datos json sobre SQLite <sup>31</sup> o Mysql <sup>32</sup>, no se realiza de manera nativa sino mediante paquetes de terceros.

**Matlab, Versión: 2015b**

El motor de cálculo usado en esta aplicación ha sido Matlab. Este requisito ha sido uno de los principales condicionantes de la selección tecnológica, ya que el MAE [4] ya estaba implementado bajo este lenguaje. Matlab cuenta con una API<sup>33</sup> que permite poder ser instalado como un paquete en Python, y de esta manera usarlo como una aplicación más, invocando al entorno y rutinas del propio software.

**Plotly, Versión: 2.0.2**

Para la representación de gráficos se ha escogido Plotly, que cuenta con una API que permite instalar la aplicación como un paquete más en Python. Y de esta forma poder integrar la librería de gráficos interactivos dentro de la aplicación.

**Bootstrap, Versión: 3.3.7**

Este framework css ha permitido implementar la interfaz web de una manera rápida y sencilla. Se ajusta perfectamente al diseño planteado para la interfaz de usuario, es posible su uso dentro de la herencia de plantillas que usa Django y va a permitir responder a uno de los requisitos de una manera fácil. RQNF -3 Diseño de una interfaz *responsive*.

**5.3. Arquitectura de la aplicación****5.3.1.1. Implementación del patrón Modelo Vista Template**

Como se vio en el apartado **2.3.2.1 Patrón Modelo Vista Controlador**, ciertos aspectos de este patrón han sido modificados, dando paso a variantes del concepto original. Aunque **Django** sigue de manera cercana el patrón MVC, éste desarrolla su propia implementación. Con Django el controlador es gestionado por el propio framework, por lo que donde realmente se centra el desarrollo es: en los modelos, las plantillas (templates) y las

---

<sup>31</sup> <https://github.com/dmkoch/django-jsonfield>

<sup>32</sup> [https://django-mysql.readthedocs.io/en/latest/model\\_fields/json\\_field.html](https://django-mysql.readthedocs.io/en/latest/model_fields/json_field.html)

<sup>33</sup> Matlab API para otros lenguajes: <https://es.mathworks.com/help/matlab/programming-interfaces-for-c-c-fortran-com.html>

vistas, es por esto que se suele hablar de Django como un framework Modelo Vista Template (**MVT**). En la Figura 39, se puede ver un diagrama del funcionamiento del MTV de Django:

- 1: El navegador manda una solicitud.
- 2: La **vista** interactúa con el **modelo** para obtener los datos.
- 3: La **vista** llama a la **plantilla**.
- 4: La **plantilla** renderiza la respuesta a la solicitud del navegador.

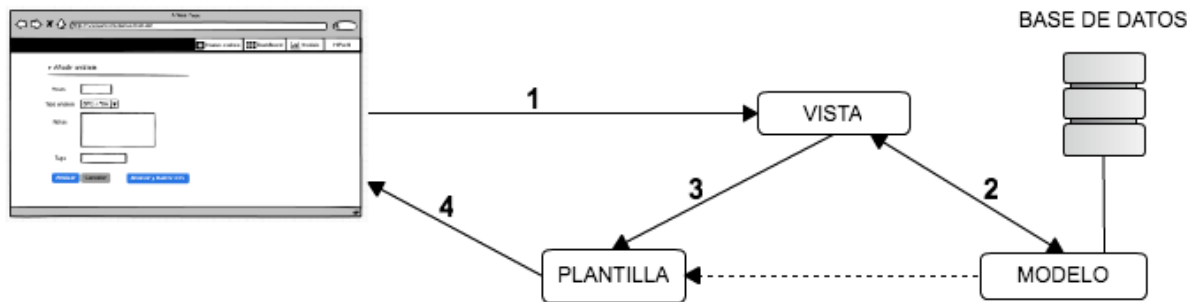


Figura 39. Funcionamiento del MTV de Django

En la siguiente tabla, Tabla 43, se puede ver la analogía entre el patrón MVC y el MTV de Django, en donde la diferencia más sustancial es que en la vista de del patrón MTV reside la lógica del negocio, y el *template* o plantilla se encarga de la capa de presentación.

Capa Lógica	MVC	MTV
Acceso a base de datos	Modelo	Modelo
Presentación	Vista	Template
Negocios	Controlador	Vista

Tabla 43. Analogía entre MVC y MTV

### 5.3.1.2. Arquitectura Django

Una aplicación web, normalmente está a la espera de recibir solicitudes HTTP desde un navegador web. Cuando se recibe la solicitud la aplicación la resuelve en base a la url y a la información solicitada, dependiendo de si es necesario leer (get) o escribir (post) en la base de datos o realizar otra serie de acciones. La respuesta generada por la aplicación por lo general devuelve una página html, para que se muestre en el navegador los datos solicitados. En el siguiente diagrama, Figura 40, se puede ver como una aplicación web basada en Django realiza estos pasos.

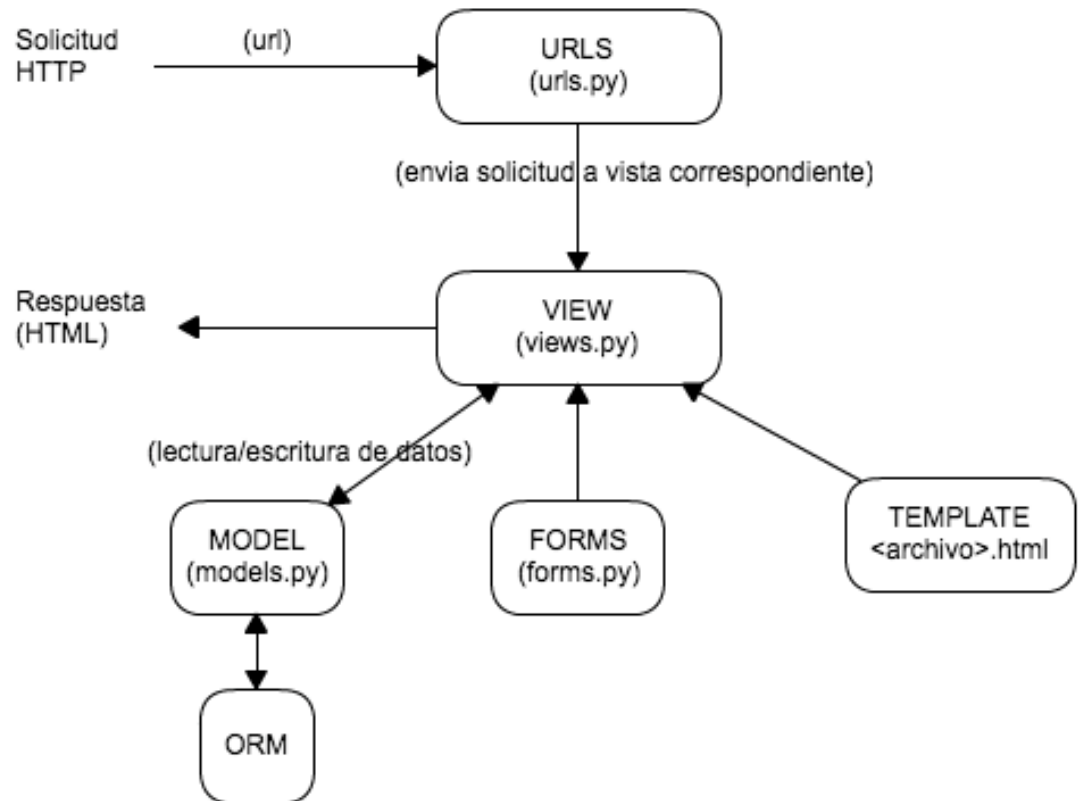


Figura 40. Diagrama del flujo de una solicitud en Django

La implementación de Django permite estructurar el código y la lógica de la aplicación en archivos independientes. URLs, VIEW, MODELS, FORMS y TEMPLATES o plantillas. A continuación, se va a describir la estructura empleada en el desarrollo, junto con el contenido de cada archivo. El código que contiene se puede consultar en el repositorio del proyecto <sup>34</sup>. Django permite que sobre un mismo proyecto se ejecuten varias aplicaciones. Para este desarrollo se planteó un proyecto principal, llamado **WEBSITE**, y una aplicación, llamada **AEPRO**. De esta manera se ha intentado separar la parte que corresponde al *site*, como pueda ser la gestión de usuarios, de la parte que es propia de la aplicación, como cálculos y representación de gráficos.

Mediante los siguientes diagramas, Figura 41 y Figura 42, se intenta explicar y simplificar los ficheros que contiene cada una de las partes y la relación que guardan entre ellos. Nótese que en cada uno de los ficheros se hace alusión a que contienen algún tipo de función *methodA()*, *methodB()*, en referencia a las funciones y clases que contienen. Como se ha

<sup>34</sup> <https://github.com/merclsb/AEPRO>

comentado anteriormente el contenido de los mismos puede ser consultado en el repositorio del proyecto.

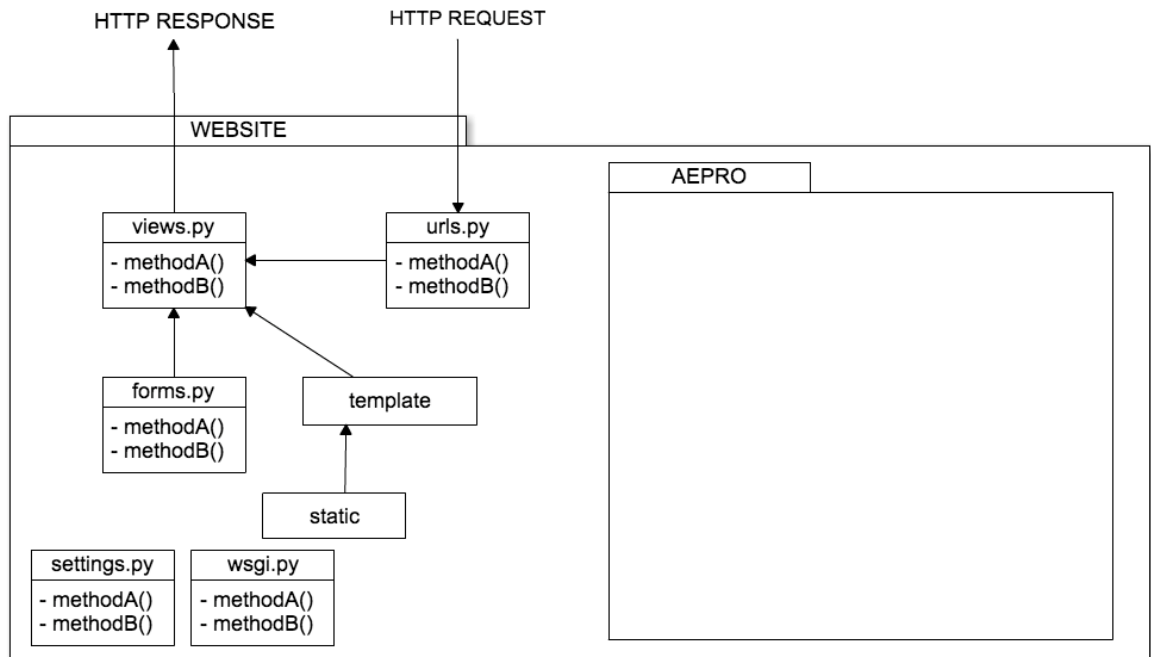


Figura 41. Diagrama de interacción empleado para el proyecto principal (WEBSITE)

A continuación, se describe a grandes rasgos el contenido de cada uno de los ficheros que componen la estructura del proyecto principal **WEBSITE**:

### **urls.py**

Se trata del mapeador de urls cuya función es redirigir las solicitudes http a la vista apropiada, en función de la urls solicitada. El **urls.py** de WEBSITE contiene todas las urls del proyecto, y será desde el cual se gestione si es una petición para contenido de WEBSITE o para la aplicación AEPRO.

### **views.py**

Contiene las funciones necesarias para la gestión del perfil de usuarios.

### **forms.py**

Contiene los formularios necesarios para la gestión de los perfiles de usuarios.

### **settings.py**

Se trata del archivo de configuraciones del proyecto, donde se indica que módulos se emplean, cuales son las rutas del proyecto, la conexión a la base de datos, etc.

**static**

Directorio en cual se guardan todos los archivos estáticos que va a necesitar la aplicación: javascript, css, fonts e imágenes

**template**

Este directorio contiene todas las plantillas (archivos .html) necesarios para la construcción de la aplicación. Por un lado, se encuentran los necesarios para la construcción de la estructura:

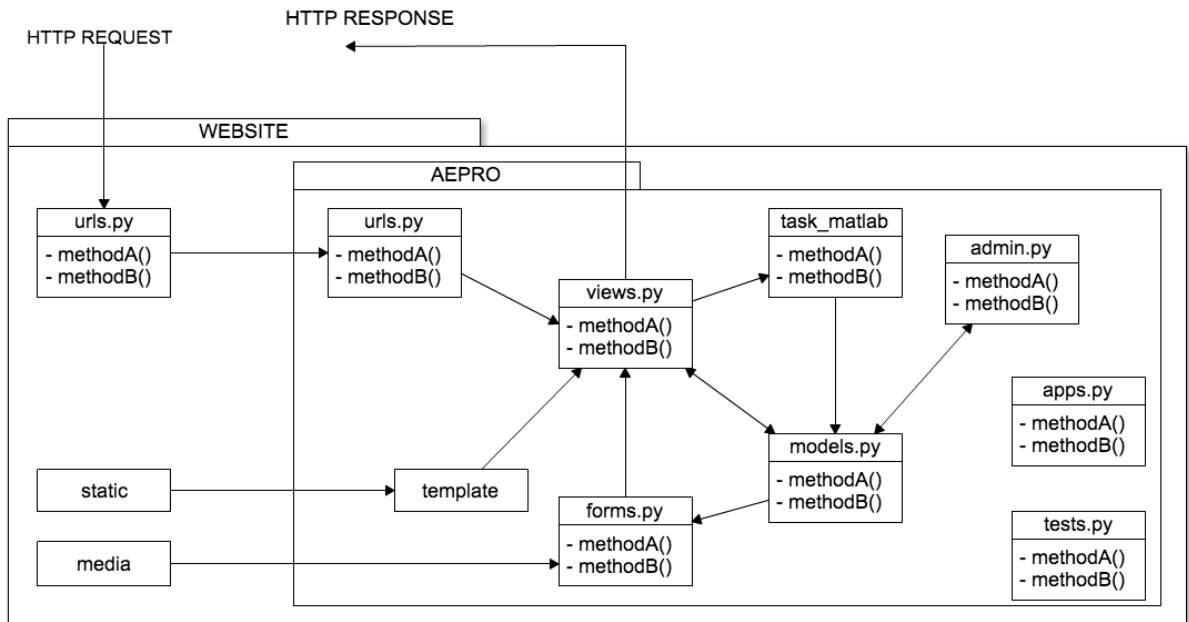


Figura 42. Diagrama de interacción empleado para la aplicación (AEPRO)

Como puede verse en el diagrama, el urls principal apunta al urls de la aplicación AEPRO, para poder solicitar la vista necesaria. La parte central de proyecto y en la que está contenida casi toda la lógica de la aplicación son tres archivos:

**Forms.py :**

Se trata de un punto crítico del sistema ya que es a través de los formularios como se va a proceder a dotar al sistema de datos. Es por tanto el punto en el que hay que realizar las validaciones y comprobaciones, así como aplicar las restricciones necesarias: comprobar que los campos están cumplimentados y sean del tipo de datos esperado, que las fechas introducidas lleven un orden lógico, comprobar el tipo de análisis, el periodo a analizar y que las fechas se ajusten al tipo de análisis, que el fichero proporcionado sea de tipo xls y comprobar los campos fecha y numéricos.

**Views.py**

Contiene todas las clases y funciones necesarias para implementar la lógica de la aplicación, accediendo a los datos necesarios para satisfacer las solicitudes y pasan esta información a las plantillas para su representación.

Para cada nuevo análisis se genera un subproceso que invoca a `task_matlab` y le pasa los datos necesarios para proceder a crear una nueva instancia.

Las funciones encargadas de generar las representaciones de los resultados, es decir las gráficas hacen uso de Plotly para generar las imágenes que se pasan a las plantillas para su representación.

**Task\_matlab.py**

Se consideró desligar la tarea interna de comunicarse con la API de Matlab del resto de la lógica implementada en el fichero `views.py`. En este fichero se habilita el entorno para poder trabajar con orm de Django, se prepara la información para lanzar instancias de Matlab y se procesa los resultados para poder almacenarlos en la BD.

### 5.4. Modelo de datos

En la Figura 43 se muestra el diagrama de datos obtenidos para la base de datos. Se puede observar cómo además de los propios generados a partir de los modelos, Django implementa los necesarios para el servidor.

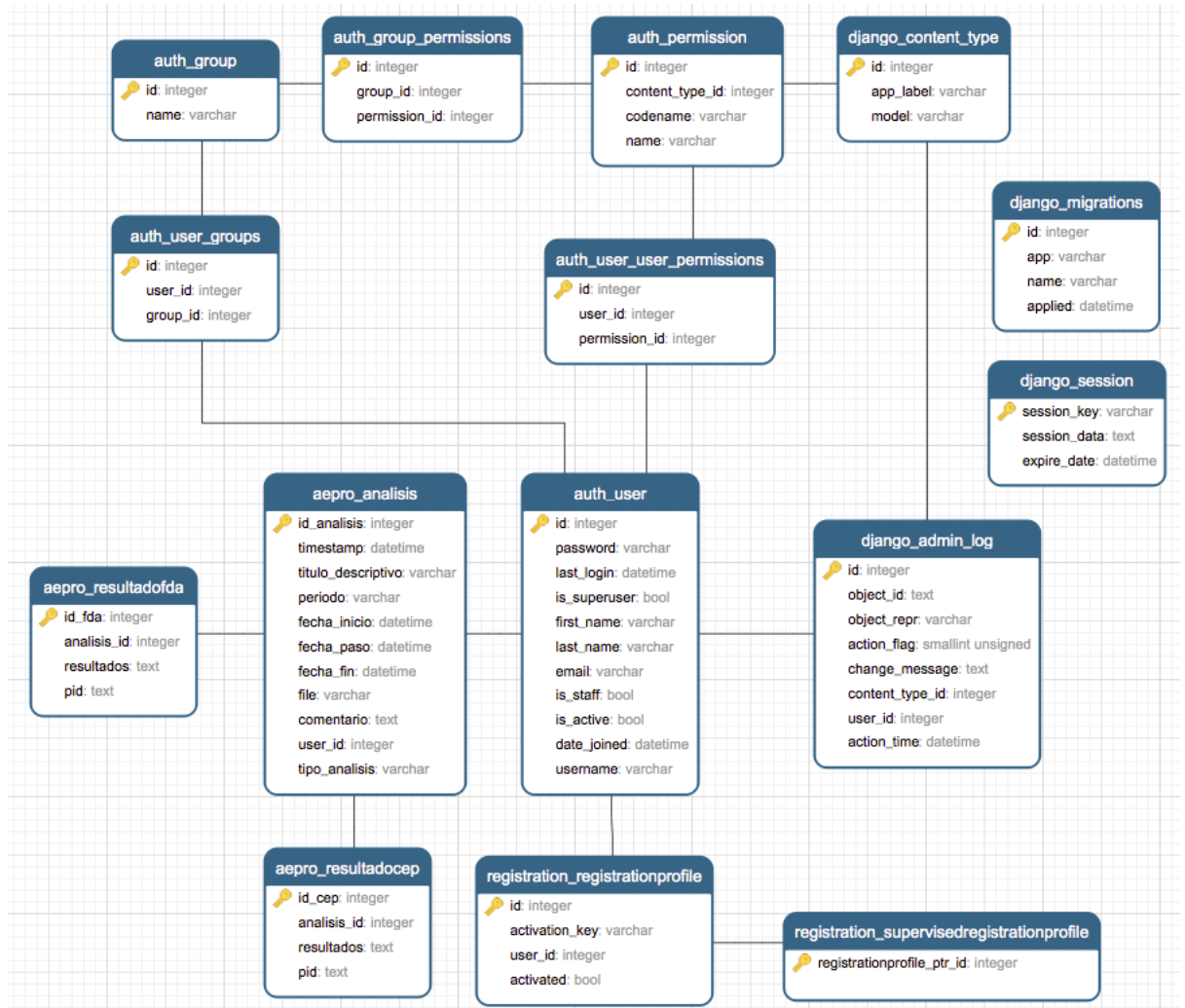


Figura 43. Diagrama del modelo de datos de la aplicación.

## 5.5. Implementación de la interfaz de usuario

En este apartado se va a presentar el aspecto que muestra la aplicación una vez implementado, así como su relación con los diseños planteados y los requisitos. En el Anexo I. Manual de usuario, se pueden ver las distintas secciones de la aplicación.

La siguiente vista, Figura 44, muestra la funcionalidad de la aplicación para crear un nuevo análisis, en relación con el diseño planteado en la Figura 31. En este formulario el usuario debe cumplimentar los campos con la información necesaria y seleccionar el tipo de análisis a realizar, los periodos a analizar, y elegir un archivo que contendrá la información necesaria para analizar.

The screenshot shows the 'Crear Análisis' form with the following details:

- AEPRO** logo and navigation menu: + Nuevo Análisis, Resultados, Validar fichero, paco
- + Crear Análisis** header
- Título:** NO2 dublin
- Selecciona Periodo:** Mes
- Tipo analisis:**
  - Control Estadístico de Procesos
  - Analisis Funcional de Datos
- Fecha inicio:** 2013-01-01 00:00
- Fecha paso:** 2013-01-01 00:00
- Fecha fin:** 2013-12-31 00:00
- Indica un comentario si es necesario:** Escribe aquí tu comentario
- File:** Choose File, No file chosen. Solo se permite subir archivos en formato XLS!!!!
- Buttons:** Calcular, Reset
- Calendar (December 2013):**

Lu	Ma	Mi	Ju	Vi	Sá	Do
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Figura 44. Vista del apartado: crear nuevo análisis.

La siguiente vista, Figura 45, muestra la funcionalidad de la aplicación para visualizar el listado de los resultados correspondientes a los análisis ya creados, en relación con el diseño planteado en la Figura 32. Desde este listado el usuario puede ver la información relativa a los análisis creados previamente, así como el estado de los mismos: si se están procesando, si los resultados ya están disponibles o se ha producido algún tipo de error.

Titulo	Tipo Analisis	Fecha	Estado
NO2 dublin	Control Estadistico de Procesos	1 de Septiembre de 2017 a las 11:12	Pendiente de procesar...
Turbidez Ebro	Analisis Funcional de Datos	30 de Agosto de 2017 a las 13:54	Resultados FDA
Turbidez Ebro	Analisis Funcional de Datos	30 de Agosto de 2017 a las 13:44	Resultados FDA
Turbidez Ebro	Analisis Funcional de Datos	30 de Agosto de 2017 a las 13:41	Resultados FDA
Turbidez Ebro	Analisis Funcional de Datos	30 de Agosto de 2017 a las 13:40	Pendiente de procesar...
Turbidez Ebro	Analisis Funcional de Datos	30 de Agosto de 2017 a las 13:38	Pendiente de procesar...

Figura 45. Vista del apartado: lista de resultado.

En esta tercera vista, Figura 46, se puede ver el resultado obtenido para un análisis del tipo FDA, atendiendo a la propuesta de diseño planteada en la Figura 33.

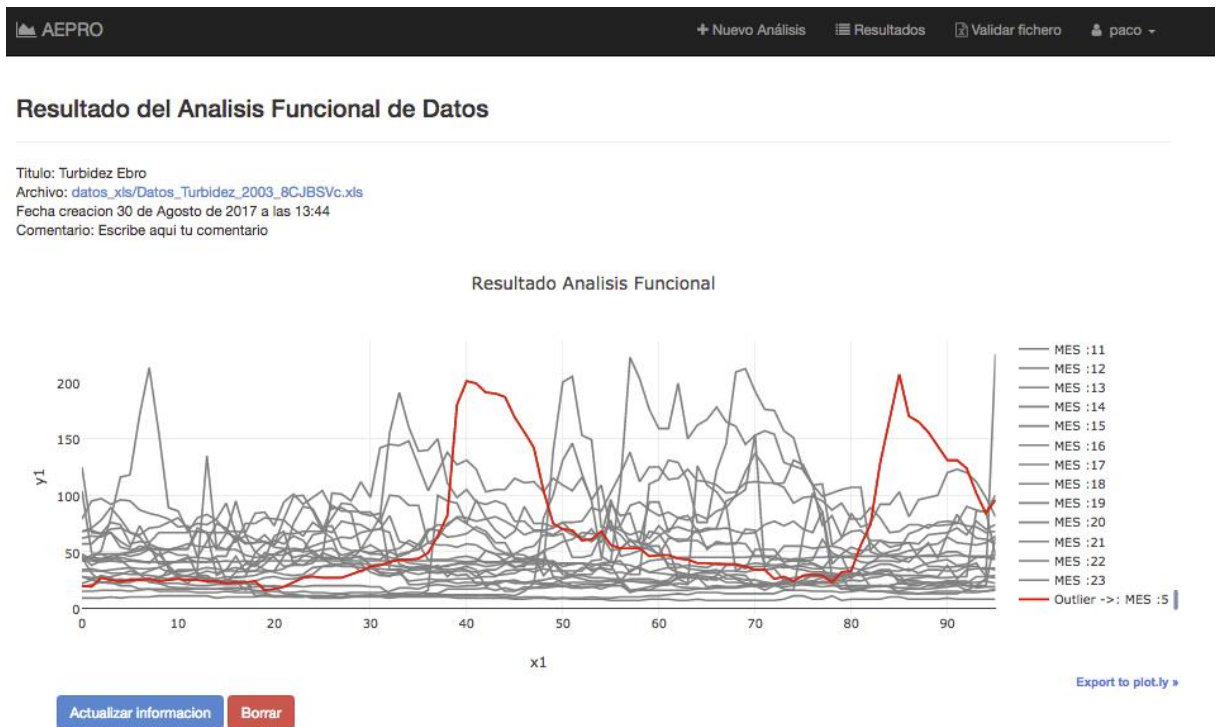


Figura 46. Vista del apartado: resultado de análisis funcional de datos.

Mediante la siguiente imagen, Figura 47, se da respuesta a la necesidad de que la aplicación pueda adaptarse a distintos dispositivos, permitiendo una correcta visualización de una misma página desde distintos dispositivos, ya sean ordenadores, tablets o móviles

Haciendo uso de la herramienta *Responsive Design View*<sup>35</sup> se puede ajustar la resolución a la que se desea que se vea nuestra aplicación. Se puede ver como el menú superior pasa a estar contenido dentro de un botón que es desplegable, y así facilitar su accesibilidad. De igual manera los campos del formulario, del apartado nuevo análisis, se ajustan al tamaño de la pantalla sin ningún problema.

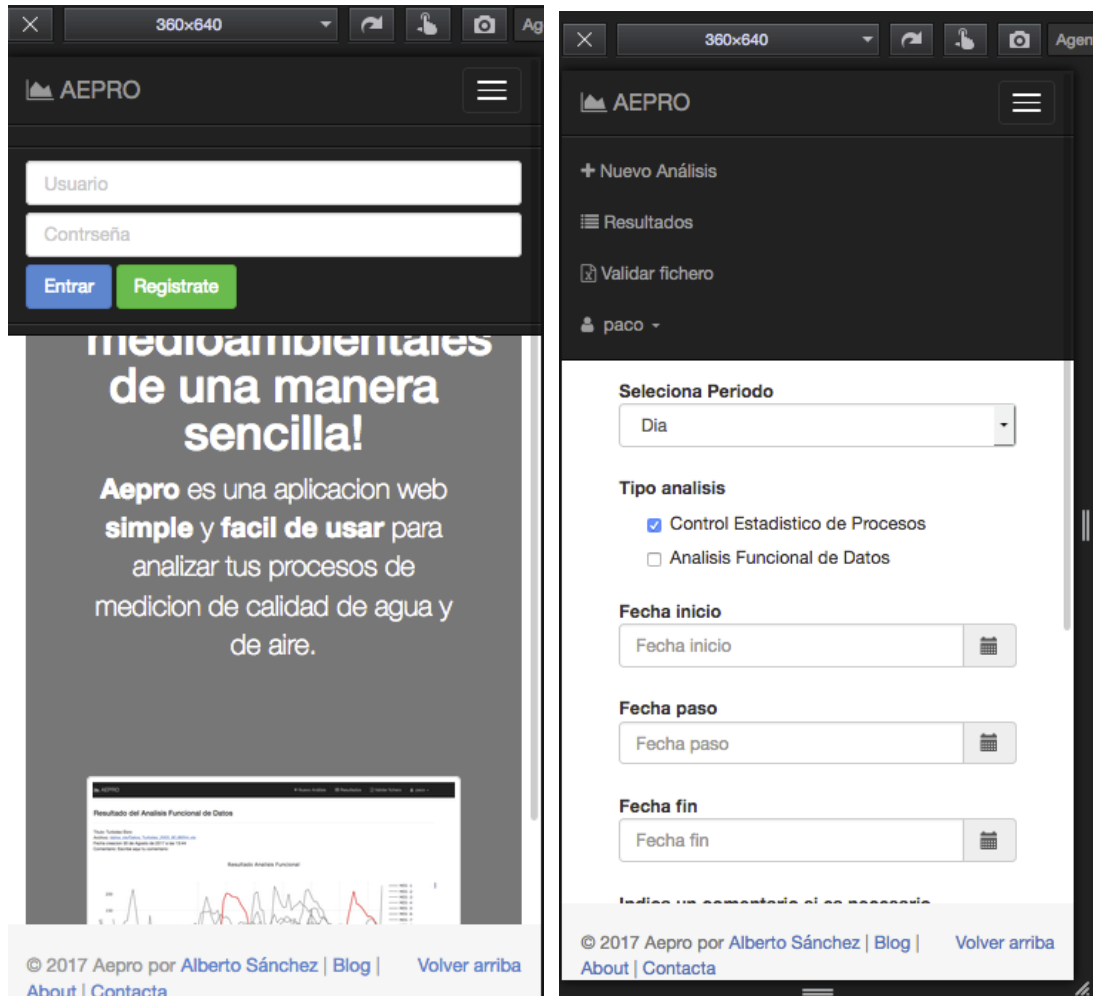


Figura 47. Vista del apartado. Diseño *responsive* o adaptativo de la aplicación.

<sup>35</sup> *Responsive Design View* disponible en:

[https://developer.mozilla.org/es/docs/Tools/Responsive\\_Design\\_View](https://developer.mozilla.org/es/docs/Tools/Responsive_Design_View)

## 6. Validación técnica de la solución

Para realizar una comprobación del correcto funcionamiento de la aplicación, poder detectar errores, así como posibles mejoras, se van a realizar dos tipos de pruebas. Pruebas del funcionamiento ordinario de la aplicación y unas pruebas de aceptación realizadas por varios usuarios.

### 6.1. Pruebas de cada subsistema

Este tipo de pruebas permite comprobar la correcta ejecución de la aplicación. Consiste en comprobar que el desempeño de la aplicación es correcto en todos sus aspectos, obteniendo las respuestas deseadas para cada acción.

Para validar el correcto funcionamiento de toda la aplicación, se ha optado por mostrar cómo se ha probado cada uno de los subsistemas que componen la aplicación: los subsistemas que componen la aplicación son:

- Subsistema alta de nuevos usuarios
- Subsistema acceso a la aplicación
- Subsistema gestión del perfil de usuario
- Subsistema creación de un nuevo análisis
- Subsistema lista de resultados
- Subsistema validación de ficheros

Las pruebas de cada subsistema pueden consultarse en el Anexo III. Pruebas de subsistema.

### 6.2. Pruebas de aceptación

El objetivo de este apartado es definir un conjunto de pruebas que deben ser ejecutadas por los usuarios para validar si el sistema cumple con los requisitos de funcionamiento esperado y procede así a la aceptación del sistema.

En las siguientes tablas, Tabla 44, Tabla 45, Tabla 46 y Tabla 47, se han descrito en detalle cada uno de los casos de pruebas (CP) que se han identificado como necesarios para poder verificar la funcionalidad completa del sistema.

<b>CP001</b>	<b>Alta de un nuevo usuario en la aplicación</b>
<b>Descripción</b>	El usuario deberá registrarse en la aplicación, creando una cuenta de usuario y activarla para acceder a la aplicación. El tiempo de ejecución será considerado desde que accede al formulario hasta que inicia sesión en su cuenta.

<b>Prerrequisitos</b>	El usuario debe contar con una cuenta de correo válida para recibir el correo de activación de la cuenta.
<b>Pasos</b>	<p><b>Basado en el caso de uso: sistema de registro y autenticación de usuario</b> (Figura 16). Dirigirse al apartado <i>registrarse</i>. El nuevo usuario debe cumplimentar el formulario de alta en el sistema:</p> <ul style="list-style-type: none"> <li>-nombre</li> <li>-email</li> <li>-contraseña</li> </ul> <p>Cuando reciba el email:</p> <ul style="list-style-type: none"> <li>-activar la cuenta mediante un enlace que se envía a su correo</li> </ul>
<b>Resultado esperado</b>	El usuario recibe un correo con un enlace para activar la cuenta. Una vez que accede al enlace ya puede acceder a la aplicación.
<b>Resultado obtenido</b>	<p><b>Número de usuarios: 2</b></p> <p><b>Usuario 1</b> Tiempo de ejecución de la tarea: 3:30 minutos Incidencias: Sin incidencias Comentarios: En el caso de que la contraseña no sea adecuada, y tener que introducir una nueva, sería adecuado que se mantuviera guardado la información de los campos.</p> <p><b>Usuario 2</b> Tiempo de ejecución de la tarea: 1:00 minuto Incidencias: Sin incidencias Comentarios: Se aplican una serie de restricciones en la contraseña como longitud mínima o complejidad que no se indican en el registro.</p>

Tabla 44. Caso de prueba 1 – Alta de un nuevo usuario en la aplicación

<b>CP002</b>	<b>Acceso de un usuario a la aplicación</b>
<b>Descripción</b>	El usuario deberá hacer uso de sus credenciales para acceder al contenido de la aplicación.
<b>Prerrequisitos</b>	El usuario debe contar con una cuenta en el sistema, ya activada.
<b>Pasos</b>	<p><b>Basado en el caso de uso: sistema de registro y autenticación de usuarios</b> (Figura 16).</p> <ul style="list-style-type: none"> <li>-Tratar de acceder a contenido de la aplicación antes de logearse.</li> <li>-El nuevo usuario debe usar su nombre de usuario y contraseña para acceder al sistema</li> <li>-Comprobar que solo es posible acceder a la información generada por el usuario. <ul style="list-style-type: none"> <li>-perfil de usuario</li> <li>-listado de análisis generados</li> </ul> </li> <li>-Una vez comprobado debe cerrar sesión</li> </ul>

<b>Resultado esperado</b>	El usuario solo es capaz de acceder al contenido de la aplicación una vez que se ha logueado. Una vez que tiene acceso a la aplicación solo puede acceder al contenido y datos asociados a su usuario.
<b>Resultado obtenido</b>	<p><b>Número de usuarios: 2</b></p> <p><b>Usuario 1</b>  Tiempo de ejecución de la tarea: 1 minuto  Incidencias: Sin incidencias  Comentarios: Sin comentarios</p> <p><b>Usuario 2</b>  Tiempo de ejecución de la tarea: 1 minuto  Incidencias: Sin incidencias  Comentarios: Sin comentarios</p>

Tabla 45. Caso de prueba 2 – Acceso de un usuario a la aplicación.

<b>CP003</b>	<b>Creación de un nuevo análisis</b>
<b>Descripción</b>	El usuario debe crear un nuevo análisis.
<b>Prerrequisitos</b>	El usuario debe comprender cuales son los datos que va a introducir en el sistema a fin de obtener unos resultados. El usuario debe tener un fichero de registros a evaluar.
<b>Pasos</b>	<p><b>Basado en el caso de uso: sistema de análisis (Figura 15)</b></p> <p><b>El usuario acceda al apartado: nuevo análisis</b></p> <ul style="list-style-type: none"> <li>-Introduce un título descriptivo</li> <li>-Selecciona el tipo de análisis que quiere realizar: CEP o FDA</li> <li>-Introduce la fecha inicial (manualmente o mediante datepicker)</li> <li>-Introduce la fecha de paso (periodo del segundo registro)</li> <li>-Introduce la fecha final</li> <li>-Escribir un comentario</li> <li>-Seleccionar el fichero que contiene los registros a analizar selecciona: <b>calcular</b> para proceder al análisis, o <b>reset</b> si quiere borrar toda la información.</li> </ul>
<b>Resultado esperado</b>	Si toda la información es correcta el sistema debe mostrar un mensaje de creación satisfactoria y mostrar un nuevo ítem en el apartado de lista de resultados.
<b>Resultado obtenido</b>	<p><b>Número de usuarios: 2</b></p> <p><b>Usuario 1</b>  Tiempo de ejecución de la tarea: 2:52 minutos  Incidencias: Sin incidencias</p>

	<p>Comentarios: Junto al termino paso, incluir una descripción de a que se refiere.</p> <p><b>Usuario 2</b>          Tiempo de ejecución de la tarea: 40 segundos          Incidencias: Sin incidencias          Comentarios: permitir algo otro tipo de formato como csv, por ejemplo.</p>
--	---

Tabla 46. Caso de prueba 3 – Creación de un nuevo análisis.

<b>CP004</b>	<b>Lista de resultados</b>
<b>Descripción</b>	El usuario podrá revisar los resultados generados a partir de los análisis creados
<b>Prerrequisitos</b>	El usuario debe haber creado con anterioridad algún análisis.
<b>Pasos</b>	<p><b>Basado en el caso de uso: sistema de análisis (Figura 15).</b></p> <p><b>El usuario acceda al apartado: resultados</b></p> <ul style="list-style-type: none"> <li>-El usuario podrá ver el estado en el que se encuentra el/los análisis previamente creados.</li> <li>-El usuario selecciona un análisis que esté disponible para consultar sus resultados: en función del tipo de análisis seleccionado pueden ser CEP o FDA.</li> <li>-El usuario modifica la información que desee dentro del análisis</li> <li>-El usuario borra los resultados del análisis, y comprueba que no consta en la lista de resultados.</li> </ul>
<b>Resultado esperado</b>	Para el caso de proporcionar un fichero con errores, el sistema indica que existe algún error. Para el caso de proporcionar un fichero con el formato correcto el sistema indica que es correcto.
<b>Resultado obtenido</b>	<p><b>Número de usuarios: 2</b></p> <p><b>Usuario 1</b>          Tiempo de ejecución de la tarea: 1:39 minutos          Incidencias: Sin incidencias          Comentarios: No se actualiza automáticamente el estado del procesado. Muy satisfecho con la información interactiva que proporcionan las gráficas.</p> <p><b>Usuario 2</b>          Tiempo de ejecución de la tarea: 1:20 minutos          Incidencias: La página de carga de resultados CEP, le cuesta cargar todos los gráficos.          Comentarios: En caso de querer recuperar un análisis de una fecha determinada sería necesario buscarlo a mano. No existe ningún tipo de filtro.</p>

Tabla 47. Caso de prueba 4 – Lista de resultados.

<b>CP005</b>	<b>Validación del formato de ficheros</b>
<b>Descripción</b>	El usuario escogerá entre varios archivos para comprobar que el sistema comprueba el correcto formato de los mismos.
<b>Prerrequisitos</b>	El usuario debe contar con 4 o 5 ficheros de distinto tipo que permitan validar las distintas restricciones y el correcto formato del mismo.
<b>Pasos</b>	<p><b>Basado en el caso de uso: sistema de análisis</b> (Figura 15).</p> <p><b>El usuario acceda al apartado: validar fichero</b></p> <ul style="list-style-type: none"> <li>-Selecciona uno de los archivos que contiene registros</li> <li>-Espera a recibir la respuesta del sistema: <ul style="list-style-type: none"> <li>-Introduce un fichero con errores.</li> <li>-Introduce un fichero con el formato correcto.</li> </ul> </li> </ul>
<b>Resultado esperado</b>	Para el caso de proporcionar un fichero con errores, el sistema indica que existe algún error. Para el caso de proporcionar un fichero con el formato correcto el sistema indica que es correcto.
<b>Resultado obtenido</b>	<p><b>Número de usuarios: 2</b></p> <p><b>Usuario 1</b>  Tiempo de ejecución de la tarea: 1:07 minutos  Incidencias: Se han validado dos archivos, uno correcto y otro con incidencias.  Comentarios: Mientras se realiza la validación, debería aparecer algún icono que indique que el sistema está trabajando. Cuando se produce una incidencia en el archivo, debería indicar el tipo de incidencia que se ha producido.</p> <p><b>Usuario 2</b>  Tiempo de ejecución de la tarea: 1:20 minutos  Incidencias: Se han validado tres archivos, uno correcto y dos erróneos.  Comentarios: No se observa si el sistema está haciendo algún tipo de tarea. Tampoco que detalla en profundidad cual ha sido el problema encontrado.</p>

Tabla 48. Caso de prueba 5 – Validación del formato de ficheros.

## 7. Conclusiones y líneas de trabajo futuro

En este último capítulo se presentan las conclusiones sobre la problemática tratada, como se ha abordado y cuál ha sido la solución planteada. También se incluye una última sección en la cual se exponen las líneas de trabajo futuro que pueden dar un valor añadido al proyecto.

### 7.1. Conclusiones

Como ya se ha descrito a lo largo de la documentación, éste no ha sido sólo un TFG, sino que ha sido un desarrollo software analizado, diseñado e implementado a medida dentro de un proyecto real. Ésta ha sido la motivación que ha provocado cada una de las decisiones tomadas en los procesos de ciclo de vida software. Se considera que una vez realizadas las pruebas de aceptación y a la vista de lo descrito en este documento, tanto el objetivo general como los específicos planteados, al comienzo del TFG, han sido cumplidos.

Una fase fundamental al comienzo de todo proyecto, es realizar un **análisis de los entornos** existentes **para el desarrollo** del proyecto, como fue descrito en el Capítulo 2. De tal forma que una vez analizado los requisitos pueda enfocarse un diseño que facilite su posterior implementación.

Durante todo el proyecto es necesario **poner en práctica los procesos del ciclo de vida software** que permiten llevar a buen término y con garantías la finalización del proyecto. Una vez analizado los requisitos del proyecto y teniendo en cuenta el análisis de las tecnologías que eran más idóneas, se consideró que el modelo iterativo incremental, ya que era el que mejor se ajustaba a este proyecto. Permitiendo generar distintas versiones de la aplicación, en las cuales se iban añadiendo nuevas funcionalidades, publicadas en el repositorio del proyecto.

La **integración del módulo de análisis estadístico, implementado en Matlab** previamente, ha sido un requisito clave en este desarrollo. La posibilidad de trabajar con Matlab gracias a la API con la que cuenta para Python, ha sido clave para el procesamiento de la información. Como se describe en el apartado de *Arquitectura Django*, se trabajó con la API como un módulo adicional al resto de la aplicación. Su correcto funcionamiento, es decir los resultados obtenidos, quedan plasmados en los gráficos resultantes como se pueden verse en las distintas pruebas del Capítulo 6, Validación técnica de la solución.

Como en todo sistema, el control en la entrada de datos es fundamental, por ello para este proyecto se tuvo muy presente que era necesario poder procesar la información que el usuario introducía al sistema, a fin de evitar errores, contemplado durante el proceso de diseño (Diagrama de actividad Diseño de la solución) e implementación( Descripción técnica de la

solución). Para ello, se implementó una funcionalidad que permite al usuario **validar si el fichero que contiene la información bruta** a analizar presenta algún error, como puede verse en la Figura 95. Apartado de validación de formato de ficheros.

**El diseñar una interfaz agradable y práctica que permita su utilización tanto en equipos de escritorio como desde dispositivo móviles** ha sido también uno de los objetivos más importantes, ya que una aplicación no solo tiene que ser funcional, sino también visualmente agradable y usable. Durante la fase de diseño, ya se tuvo presente, Tabla 32. RQNF 3 – Diseño de interfaz adaptativa., de cara a poder contar un sistema que diese solución a este requisito.

Para ello se optó por usar Bootstrap, uno de los frameworks css más usados hoy en día, y gracias su fácil integración con Django mediante: herencia de plantillas, *template tags*, etc. Han permitido implementar una interfaz que es capaz de servir el mismo contenido independientemente del dispositivo, como puede verse en las pruebas realizadas, Figura 47. Vista del apartado. Diseño *responsive* o adaptativo de la aplicación.

La **creación de un sistema de registro de usuarios y autenticación** fue analizado y tenido muy presente durante la fase de diseño como puede verse en el diseño de la interfaz de usuario (Anexo II. Prototipos de la interfaz de usuario) y en modelado de la base de datos (Modelo de datos), así como durante la fase de implementación y validación ( Validación técnica de la solución).

El uso del framework Django ha facilitado en gran medida esta labor. Mediante el servidor que contiene es posible implementar un sistema de registro y autenticación de usuarios de una manera sencilla, elegante y robusta. Como se ha mostrado es posible, no solo darse de alta en el sistema, sino también restaurar la contraseña en caso de pérdida, o modificar información del usuario.

Para **producir como resultado del análisis, gráficos interactivos que pueden ser descargados por el usuario en su dispositivo**, se analizaron las diferentes librerías de gráficos 2D. las cuales tenían que permitir la representación de los gráficos estadísticos necesarios y permitir cierta interacción, así como la posibilidad de ser descargados. También se tuvo en consideración durante la fase de diseño, en especial el modelo de datos, ya que era muy necesario conocer el tipo de datos a almacenar para su posterior representación. Plotly fue la opción escogida para este objetivo, y a tenor de los resultados obtenidos se ha comprobado que este objetivo se ha alcanzado.

Por último, la posibilidad de **ampliar el alcance del uso de la herramienta** es posible ya que los usuarios que la utilizan no tienen la necesidad de adquirir la aplicación (Matlab) ni de instalar ningún software adicional.

## 7.2. Líneas de trabajo futuras

El objetivo de esta primera etapa del sistema (que engloba este proyecto) tenía como objetivo ser una primera versión estable de la aplicación y que se pudiera valorar su utilidad. Mediante el seguimiento de un mantenimiento perfectivo, se plantean una serie de modificaciones sobre esta primera versión del sistema resultado de las pruebas de aceptación, así como posibles nuevas funcionalidades para el sistema.

Como siguiente paso natural en todo desarrollo, se planteará el despliegue de la aplicación en un entorno de producción. Debido a los requisitos técnicos de la aplicación (principalmente Matlab) es necesario valorar las distintas plataformas *Cloud* disponibles, como, por ejemplo, *Amazon web services (AWS)*, *Azure* o *Digital Ocean* entre otros, ya que supondrá un coste económico que depende de la asignación fijada en el proyecto de investigación.

También se planteará el desarrollo de una API REST que permita obtener la información generada en los análisis, para ser utilizada por los usuarios con otros fines. De esta forma, no solo se podrá visualizar la información mediante los gráficos propuestos, si no que los usuarios podrán usarlos para otros fines. Como principal propuesta se estudiará *django-rest-framework*<sup>36</sup>, para la implementación de esta funcionalidad.

Como se ha mencionado en el apartado anterior, se implementará un panel de mando que contenga estadísticas para los administradores y que permita gestionar la aplicación: revisando análisis y resultados ya creados por los usuarios, dando acceso a los registros previos de los usuarios, etc.

También se valorará el ampliar los gráficos resultantes del análisis CEP, que permitan obtener una mayor información, como pueda ser el caso de incluir figuras de auto correlación y distribución normal, entre otras.

En cuanto las mejoras planteadas, como resultado de las pruebas de aceptación, se valoran las siguientes mejoras a implementar:

- En primer lugar, permitir el uso de un mayor número de formatos que contengan los registros, como csv, xml entre otros.
- El poder detener un análisis ya comenzado, por diversos motivos, como puedan ser una duración excesiva, una incorrecta selección de valores, etc.
- En cuanto a los gráficos resultantes del análisis CEP, tener la posibilidad de seleccionar los gráficos que se desean mostrar permitirá reducir el tiempo de carga de la página. Ya

---

<sup>36</sup> Django-Rest disponible en: <http://www.django-rest-framework.org/>

que la carga directa de todos los gráficos implica una duración elevada que penaliza la respuesta de carga de la página, lo que penaliza en la experiencia de usuario.

- Dentro de la sección de resultados incluir uno o dos filtros que permitan discriminar los resultados en función de, por ejemplo: tipo de análisis o fecha de creación.
- También, dentro de la sección resultados, la columna “estado” que se pudiera refrescar automáticamente cuando se produjera un cambio.
- Mostrar una descripción más detallada en relación a los errores ocurridos durante el proceso de análisis de la información

## Bibliografía

- [1] J. J. Pastor, J. Martínez Torres, J. J. Sancho Val, M. araujo Fernandez, y A. Sanchez Romero, «Diseño de una metodología de control de parámetros ambientales e implementación de un sistema de alerta y de protección ante desviaciones», 2015.
- [2] J. Sancho *et al.*, «Study of Water Quality in a Spanish River Based on Statistical Process Control and Functional Data Analysis», *Math. Geosci.*, vol. 48, n.º 2, pp. 163-186, 2015.
- [3] J. Ramsay, G. Hooker, y S. Graves, *Functional Data Analysis*. 2009.
- [4] T. Doctoral y J. J. Sancho Val, «Nuevas técnicas de control de calidad para estudios ambientales», 2015.
- [5] F. B. Díez, J. M. Tenías, y S. Pérez-Hoyos, «Efectos de la contaminación atmosférica sobre la salud: Una introducción», *Revista Española de Salud Pública*, vol. 73, n.º 2, pp. 109-121, 1999.
- [6] M. Oyarzún, «Contaminación aérea y sus efectos en la salud», *Rev. Chil. enfermedades Respir.*, vol. 26, pp. 16-25, 2010.
- [7] Elconfidencial, «30 medidas del Plan A contra la contaminación del Ayuntamiento de Madrid», *El Confidencial*, 2017.
- [8] JOSEP CATÀ, «Barcelona veta la circulación de los coches de más de 20 años», *El País*, 2017.
- [9] La Vanguardia, «París toma drásticas medidas por el peor episodio de contaminación de los últimos 10 años», *La Vanguardia*, 2016.
- [10] M. Katz, «Interactive web pages in MATLAB», *Matlab*, 2008. [En línea]. Disponible en: <http://blogs.mathworks.com/community/2008/02/25/interactive-web-pages-in-matlab-part-3/>.
- [11] M. Lutz, *Programming Python*, 4th ed. 2010.
- [12] J. Conallen, *Building Web Applications with UML*. Addison-Wesley, 2003.
- [13] G. Garcia Millan, «Diseño de aplicaciones web», en *Libro de la asignatura: Procesos de Ingeniería Software*, UNIVERSIDAD INTERNACIONAL DE LA RIOJA (UNIR), 2016, p. 33.
- [14] A. Holovaty y J. Kaplan-Moss, *La guía definitiva de django*. Anaya Multimedia, 2010.

- [15] R. Elmasri y S. B. Navathe, *Fundamentos de sistemas de bases de datos*, 5.<sup>a</sup> ed. 2007.
- [16] M. Keith y M. Schnicariol, «Object-Relational Mapping», en *Pro JPA 2: Mastering the Java<sup>®</sup> Persistence API*, Berkeley, CA: Apress, 2010, pp. 69-106.
- [17] S. W. Ambler, «The Object-Relational Impedance Mismatch», 2017. [En línea]. Disponible en: <http://www.agiledata.org/essays/impedanceMismatch.html>.
- [18] J. O. Meiert, *The little book of html/css framwroks*. O'Reilly, 2015.
- [19] J. Spurlock, *Bootstrap Responsive Web Development*. O'Reilly Media, 2013.
- [20] I. Alcalde, *Visualización de la información: de los datos al conocimiento*. Barcelona, ES: Editorial UOC, 2015.
- [21] J. L. Williams, *Learning HTML5 Game Programming*: 2012.
- [22] Microsoft, «SVG vs Canvas How to choose». [En línea]. Disponible en: [https://msdn.microsoft.com/es-es/library/gg193983\(v=vs.85\).aspx](https://msdn.microsoft.com/es-es/library/gg193983(v=vs.85).aspx).
- [23] M. Bostock, V. Ogievetsky, y J. Heer, «D3; Data-Driven Documents», *IEEE Trans. Vis. Comput. Graph.*, vol. 17, n.º 12, pp. 2301-2309, dic. 2011.
- [24] Google, «Google Charts», 2015. [En línea]. Disponible en: <https://developers.google.com/chart/interactive/docs/>. [Accedido: 05-abr-2017].
- [25] S. Brunel, D. Perkins, y E. Timberg, «ChartJS», 2017. [En línea]. Disponible en: <http://www.chartjs.org/docs/latest/>.
- [26] Highcharts, «Highcharts», 2017. [En línea]. Disponible en: <https://www.highcharts.com/>.
- [27] Alex Johnson, Jack Parmer, C. Parmer, y M. Sundquist., «Plotly», 2017. [En línea]. Disponible en: <https://plot.ly/>.
- [28] IEEE 1998, «IEEE Std 830-1998», *Especificación Requisitos Softw. según el estándar IEEE 830*, p. 19, 2000.
- [29] I. Sommerville, «Ingeniería del software», *Danielr.Obolog.Es*. 2005.
- [30] R. S. PRESSMAN, *INGENIERIA DEL SOFTWARE: UN ENFOQUE PRACTICO*. S.A. MCGRAW-HILL, 2001.
- [31] R. S. Pressman, *Software Engineering A Practitioner's Approach 7th Ed - Roger S. Pressman*. 2009.
- [32] S. Chacon y B. Straub, *Pro Git*, 2.<sup>a</sup> ed. Apress, 2014.

- [33] J. O. Ramsay y H. Wickham, «Package FDA». p. 293, 2015.
- [34] J. Sancho, J. Martínez, J. J. Pastor, J. Taboada, J. I. Piñeiro, y P. J. García-Nieto, «New methodology to determine air quality in urban areas based on runs rules for functional data», *Atmos. Environ.*, vol. 83, pp. 185-192, 2014.

## Anexo I. Manual de usuario

Se va a aprovechar este apartado para realizar un breve manual de usuario explicando las distintas secciones de la aplicación.

### Landing page

La siguiente Figura 52, muestra el aspecto que tiene la página inicial de la aplicación. Desde ésta página los usuarios pueden acceder al sistema si ya cuentan con una cuenta de usuario, o darse de alta en el apartado registrarse.



Figura 48. Landing page de la aplicación AEPRO.

## Alta de nuevo usuario

Desde pagina anterior, basta con dirigirse al apartado de registro. Rellenando los 4 campos de la imagen, Figura 50. Sobre este formulario el sistema aplica varias comprobaciones como que: el usuario o email no estén dados de alta previamente, la contraseña tenga una longitud mínima o no sea parecido el nombre de usuario a la contraseña.

**Regístrate:**

Nombre de usuario\*

Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only.

E-mail\*

Contraseña\*

Contraseña (confirmación)\*

Para verificar, introduzca la misma contraseña anterior.

Enviar

Figura 49. Página de registro para un nuevo usuario.

La Figura 50, muestra el mensaje "Te has registrado correctamente" si ha sido posible dar de alta al usuario.

Te has registrado correctamente. Te hemos enviado un email con un enlace para activar tu cuenta.

Figura 50. Mensaje de registro correcto para el usuario.

Y recibe un email, como el de la siguiente Figura 51, en su cuenta de correo con el link que le permite activar su cuenta AEPRO.



Figura 51. Email enviado al usuario para la activación de la cuenta.

## Página de inicio

Esta primera página, Figura 52, se usa para introducir al usuario en el contexto y funcionalidad de la aplicación.

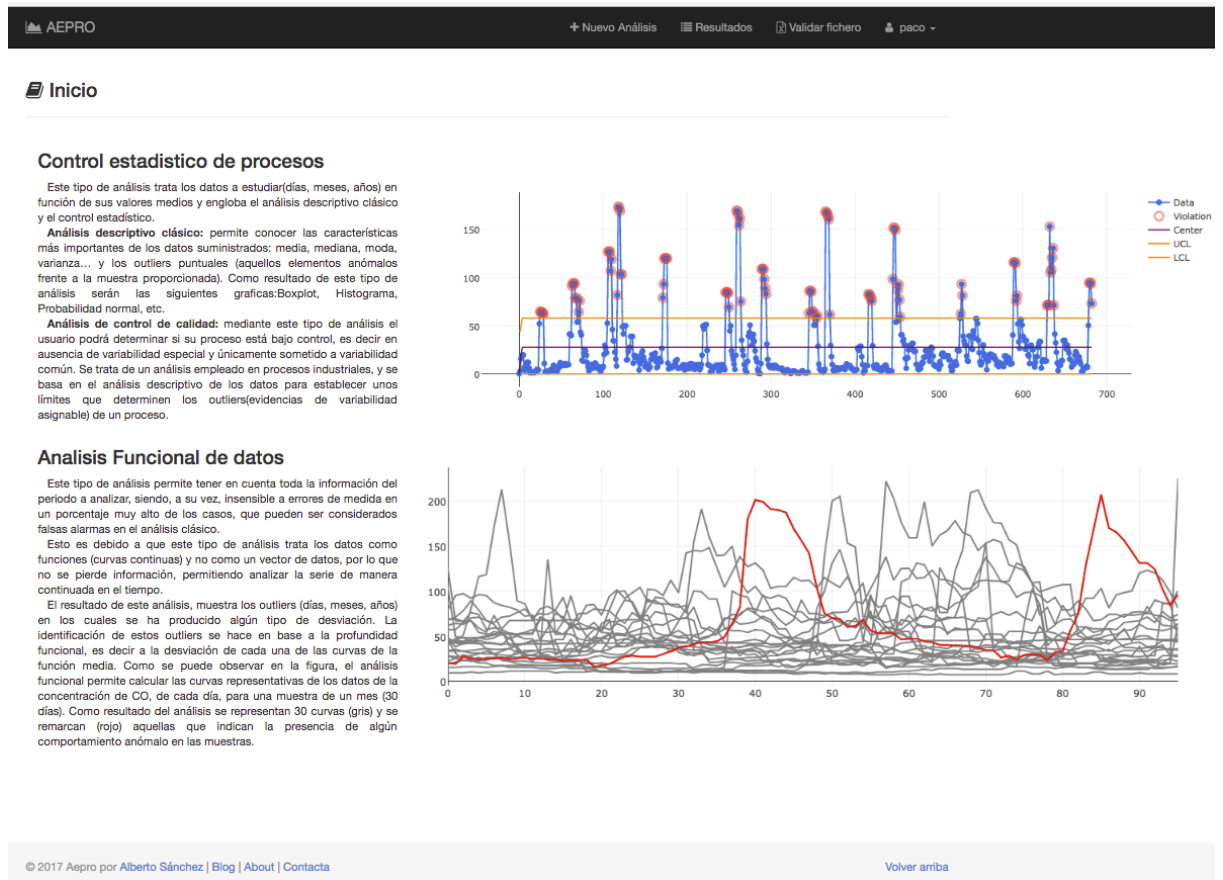


Figura 52. Página de inicio de la aplicación AEPRO.

### Crear un nuevo análisis

En el apartado **Nuevo análisis**, el usuario podrá crear un análisis que le permita procesar su información. Para ello deberá completar la información, como el ejemplo de la Figura 53. Para ello debe:

- Indicar un título descriptivo.
- Seleccionar el periodo temporal sobre el que analizar la información: días meses o años
- Seleccionar el tipo de análisis. Puede seleccionarse el tipo CDP el tipo FDA o ambos. De esta manera se realizan ambos tipos de análisis para el mismo periodo temporal.
- Indicar las fechas en las que se tomaron los registros:
- Fecha inicial: primera fecha que se registró.
- Fecha de paso: diferencia entre el primer registro y el segundo(1 minuto, 15, 30, una hora , un día, etc)
- Fecha final: fecha en que se tomó el último registro.
- Opcionalmente un comentario
- El fichero que contiene todos los registros. Sobre este fichero se realizan una serie de comprobación como: que el sea del tipo correcto (xls), que tenga una estructura correcta, o que tipos de datos sean correcto.

Completado el formulario y comprobado que todos los campos necesarios están completos y son correctos, el sistema mostrará el mensaje de cracion correcta, Figura 54.

**AEPRO** + Nuevo Análisis Resultados Validar fichero paco -

### + Crear Análisis

**Título** NO2 dublin

**Selecciona Periodo** Mes

**Tipo analisis**  
 Control Estadístico de Procesos  
 Analisis Funcional de Datos

**Fecha inicio** 2013-01-01 00:00

**Fecha paso** 2013-01-01 00:00

**Fecha fin** 2013-12-31 00:00

**Indica un comentario si es necesario**  
 Escribe aqui tu comentario

**File** Choose File No file chosen  
 Solo se permite subir archivos en formato XLS!!!!

Calcular Reset

« diciembre 2013 »

Lu	Ma	Mi	Ju	Vi	Sá	Do
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Figura 53. Creación de nuevo análisis con la información aportada por el usuario.

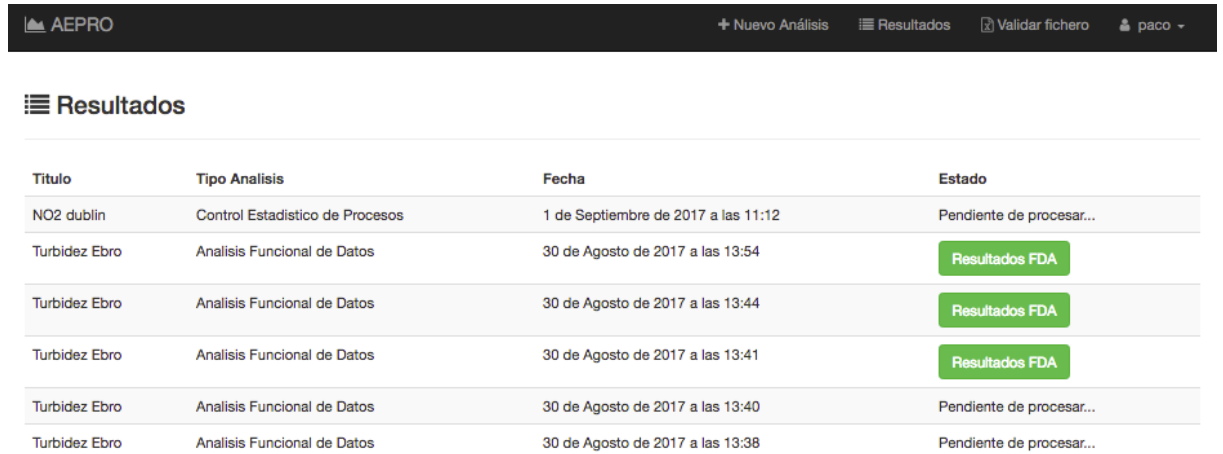
**AEPRO** + Nuevo Análisis Resultados Validar fichero paco -

**"Análisis creado correctamente."**

Figura 54. Mensaje de creación correcta de un nuevo análisis.

## Lista de resultados

Una vez creado el nuevo análisis, en el apartado de **resultados** se podrá ver cómo ha sido añadido una nueva entrada en la lista de resultados. Desde este listado se pueden ver el historial de análisis ya creados y acceder a los resultados cuando estén disponibles.



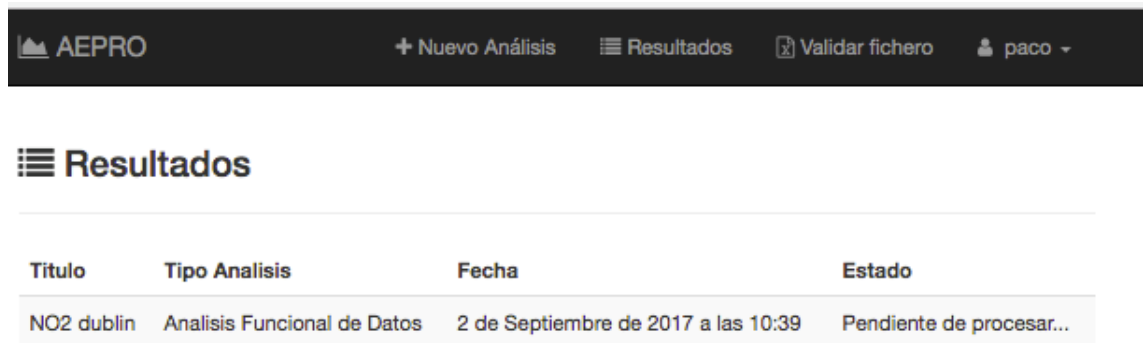
Titulo	Tipo Analisis	Fecha	Estado
NO2 dublin	Control Estadístico de Procesos	1 de Septiembre de 2017 a las 11:12	Pendiente de procesar...
Turbidez Ebro	Analisis Funcional de Datos	30 de Agosto de 2017 a las 13:54	Resultados FDA
Turbidez Ebro	Analisis Funcional de Datos	30 de Agosto de 2017 a las 13:44	Resultados FDA
Turbidez Ebro	Analisis Funcional de Datos	30 de Agosto de 2017 a las 13:41	Resultados FDA
Turbidez Ebro	Analisis Funcional de Datos	30 de Agosto de 2017 a las 13:40	Pendiente de procesar...
Turbidez Ebro	Analisis Funcional de Datos	30 de Agosto de 2017 a las 13:38	Pendiente de procesar...

Figura 55. Lista de análisis creados y el estado de procesamiento.

En las siguientes imágenes(

Figura 56, Figura 57, Figura 58, Figura 59) se muestran como el estado del análisis va cambiando en función de la fase en la que se encuentre.

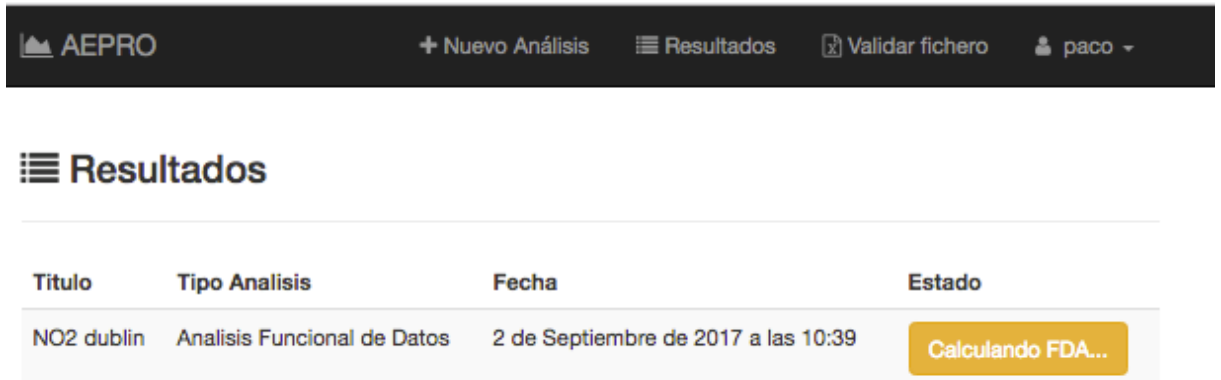
En una primera fase, el nuevo análisis se encuentra “pendiente de procesar”, es decir que está a la espera de que la información sea procesada por Matlab.



Titulo	Tipo Analisis	Fecha	Estado
NO2 dublin	Analisis Funcional de Datos	2 de Septiembre de 2017 a las 10:39	Pendiente de procesar...

Figura 56. Lista de resultados: análisis pendiente de procesar.

En una segunda fase, cuando ya está siendo calculando, se indica mediante el correspondiente estado: “Calculando tipo\_analisis”



The screenshot shows the top navigation bar of the AEPRO application with the following items: a home icon, 'AEPRO', '+ Nuevo Análisis', 'Resultados', 'Validar fichero', and a user profile 'paco'. Below the navigation bar is a section titled 'Resultados'. It contains a table with the following data:

Titulo	Tipo Analisis	Fecha	Estado
NO2 dublin	Analisis Funcional de Datos	2 de Septiembre de 2017 a las 10:39	Calculando FDA...

Figura 57. Lista de resultados: análisis calculando.

Y por último cuando ya ha sido procesada la información y se encuentra disponible se representa mediante: “Resultado tipo\_analisis” y desde el mismo botón se puede acceder a la información resultante, Figura 61, Figura 62, Figura 65 .

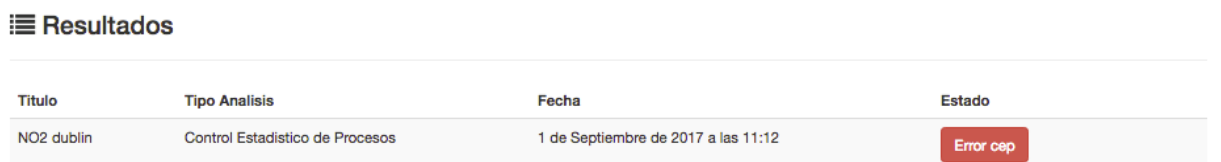


The screenshot shows the top navigation bar of the AEPRO application with the following items: a home icon, 'AEPRO', '+ Nuevo Análisis', 'Resultados', 'Validar fichero', and a user profile 'paco'. Below the navigation bar is a section titled 'Resultados'. It contains a table with the following data:

Titulo	Tipo Analisis	Fecha	Estado
NO2 dublin	Analisis Funcional de Datos	2 de Septiembre de 2017 a las 10:39	Resultados FDA

Figura 58. Lista de resultados: análisis resultado disponible.

En el supuesto de producirse alguno error durante el proceso de procesamiento de la información, se muestra al usuario el estado “error”



The screenshot shows the top navigation bar of the AEPRO application with the following items: a home icon, 'AEPRO', '+ Nuevo Análisis', 'Resultados', 'Validar fichero', and a user profile 'paco'. Below the navigation bar is a section titled 'Resultados'. It contains a table with the following data:

Titulo	Tipo Analisis	Fecha	Estado
NO2 dublin	Control Estadístico de Procesos	1 de Septiembre de 2017 a las 11:12	Error cep

Figura 59. Lista de resultados: error durante el proceso de análisis.

El usuario podrá acceder al error y borrar el análisis creado.

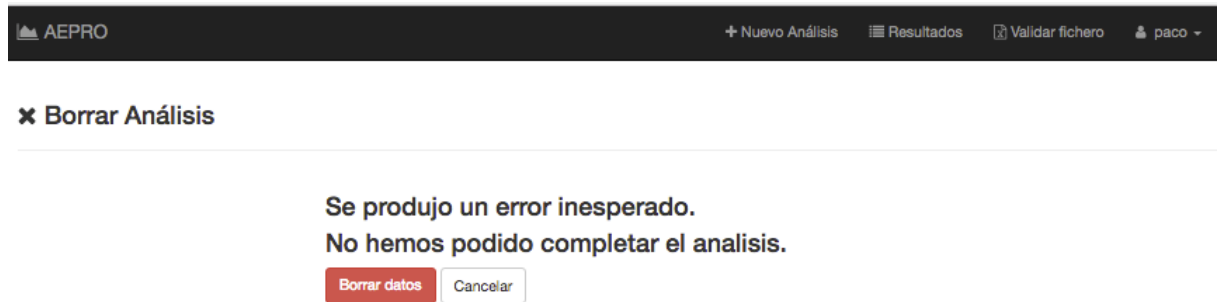


Figura 60. Mensaje previo al borrado de la información correspondiente a un error durante el proceso de análisis.

### Resultado de análisis FDA

Una vez que ya ha finalizado el análisis y es posible acceder a los resultados, se puede observar en la parte superior la información de creación del análisis y en la parte inferior los gráficos resultantes. Respecto la información correspondiente al análisis se muestra el título, el archivo que proporciono el usuario y que puede volver a descargarse, así como los comentarios y la fecha de creación. En la Figura 61, se puede ver el grafico resultante para un análisis de tipo FDA. Pueden verse las curvas correspondientes, en este caso a cada uno de los meses en color gris. Mientras que en color rojo es marcado aquel mes que se ha detectado que presenta algún tipo de desviación.

El uso de gráficos interactivos, además de permitir visualizar la información, permite al usuario poder trabajar sobre los gráficos y obtener una mayor información, Figura 62. Para este tipo de análisis, es posible discriminar únicamente la función del mes en concreto y analizar los valores obtenidos individualmente. También es posible aumentar la imagen desplazarse por ella o descargarla.

Desde este apartado es posible actualizar la información de la gráfica, Figura 63, (botón actualizar información) o borrar los resultados, Figura 64, y toda la información asociada al análisis.

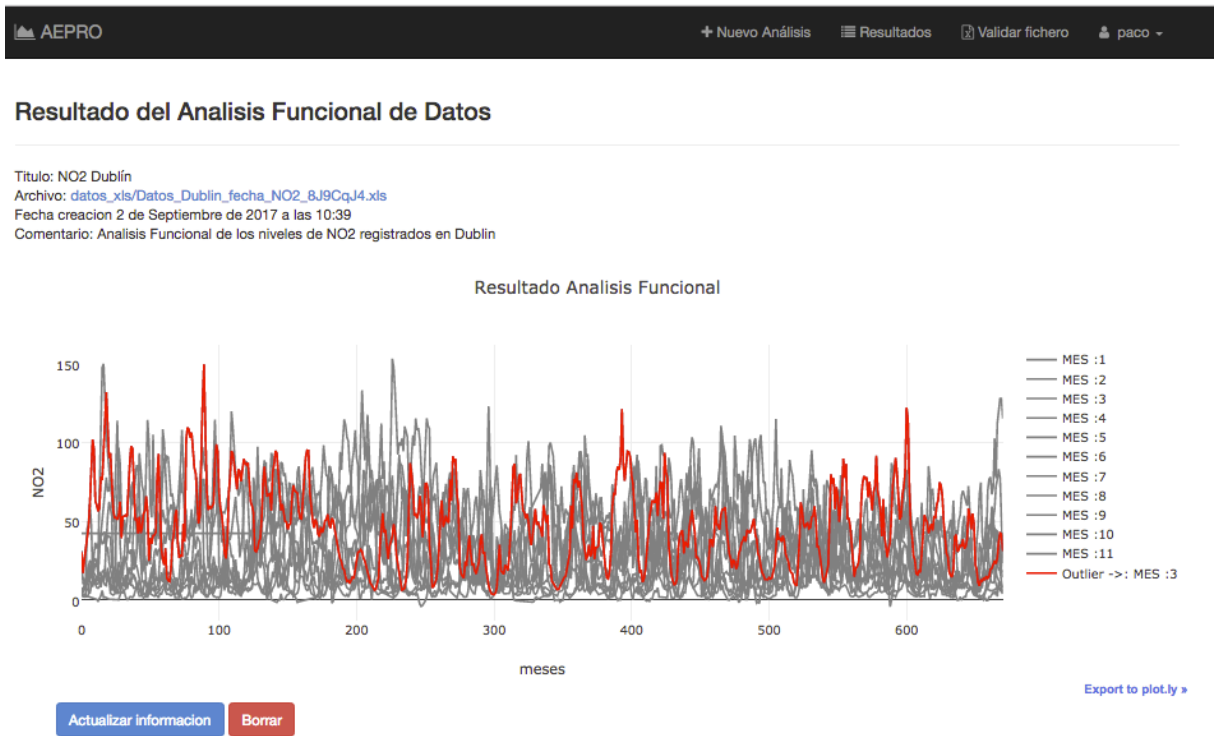


Figura 61. Resultado de un análisis FDA.

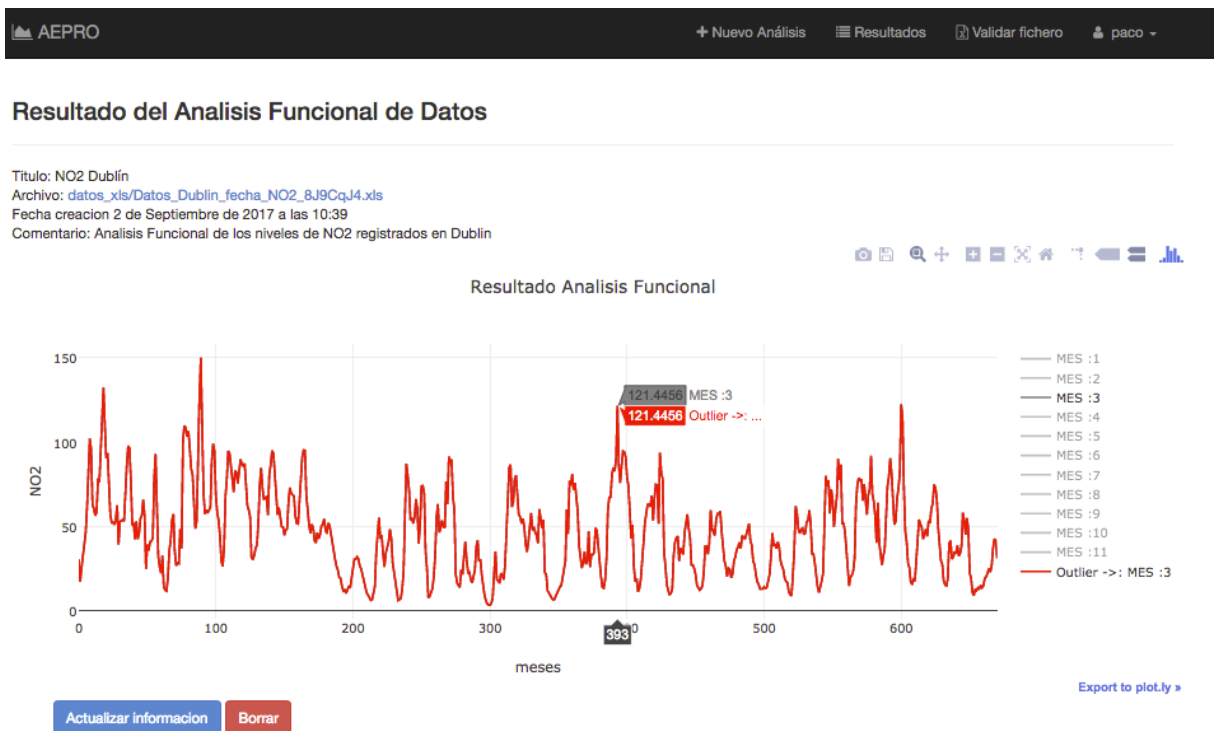
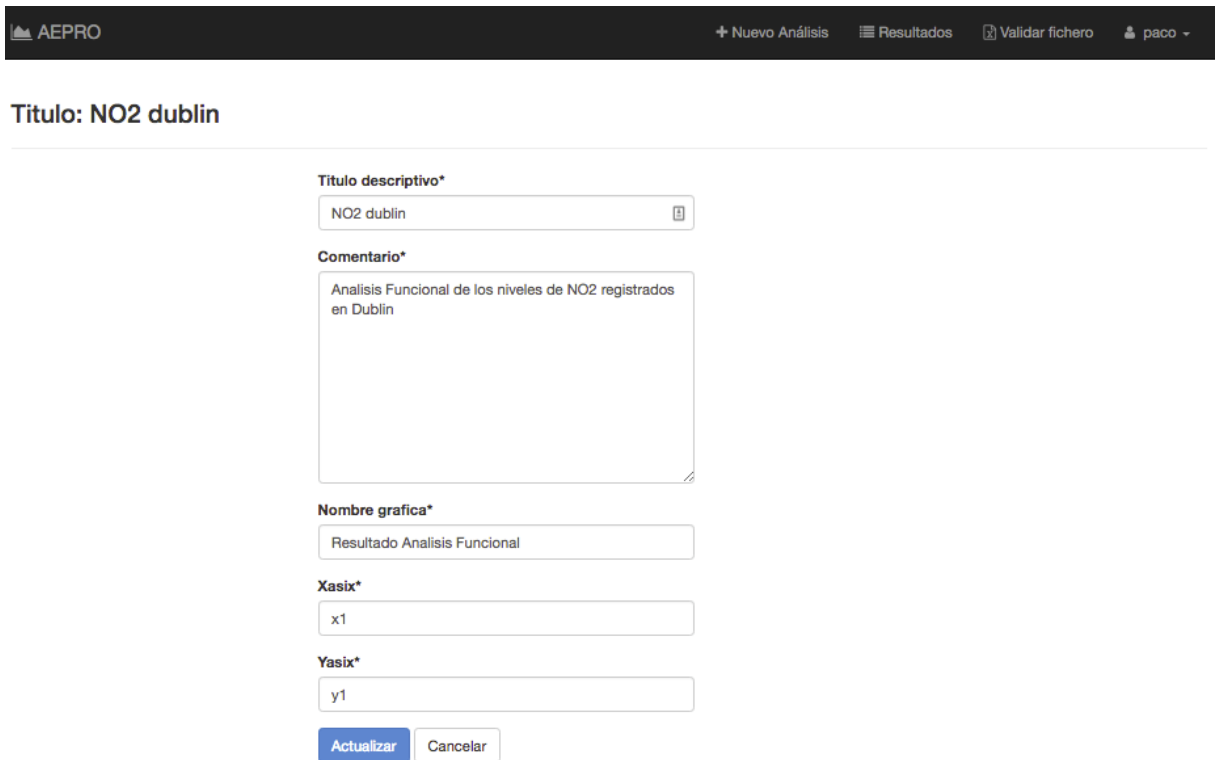


Figura 62. Resultado de un análisis FDA: manipulación de los gráficos.

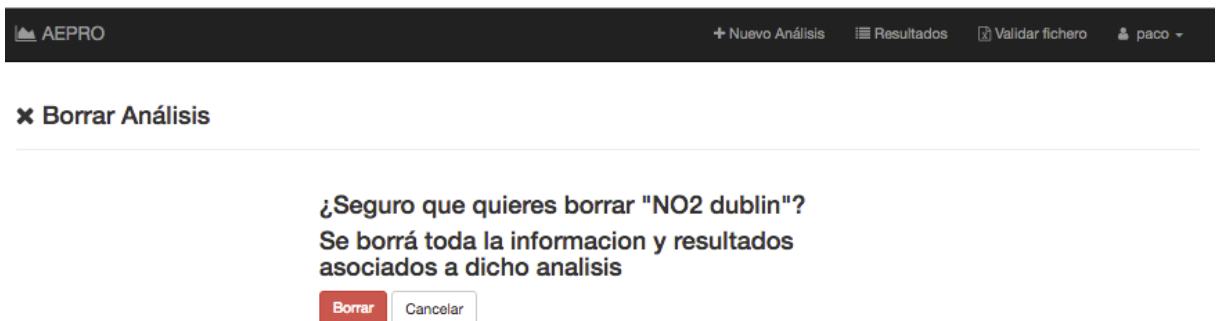


The screenshot shows the AEPRO web application interface. At the top, there is a navigation bar with the logo 'AEPRO' and several menu items: '+ Nuevo Análisis', 'Resultados', 'Validar fichero', and a user profile 'paco'. Below the navigation bar, the title of the analysis is 'Titulo: NO2 dublin'. The main content area contains a form for updating the analysis information. The form has the following fields:

- Titulo descriptivo\***: A text input field containing 'NO2 dublin'.
- Comentario\***: A text area containing 'Análisis Funcional de los niveles de NO2 registrados en Dublin'.
- Nombre grafica\***: A text input field containing 'Resultado Analisis Funcional'.
- Xaxis\***: A text input field containing 'x1'.
- Yaxis\***: A text input field containing 'y1'.

At the bottom of the form, there are two buttons: 'Actualizar' (highlighted in blue) and 'Cancelar'.

Figura 63. Apartado para actualizar la información de los resultados FDA.



The screenshot shows the AEPRO web application interface. At the top, there is a navigation bar with the logo 'AEPRO' and several menu items: '+ Nuevo Análisis', 'Resultados', 'Validar fichero', and a user profile 'paco'. Below the navigation bar, the title of the analysis is 'x Borrar Análisis'. The main content area contains a confirmation dialog with the following text:

**¿Seguro que quieres borrar "NO2 dublin"?**  
**Se borrará toda la información y resultados asociados a dicho análisis**

At the bottom of the dialog, there are two buttons: 'Borrar' (highlighted in red) and 'Cancelar'.

Figura 64. Mensaje de confirmación de borrado de análisis.

## Resultados de análisis CEP

En el caso de haber seleccionado el tipo de análisis CEP, la Figura 65 muestra los datos y gráficos resultantes. También cuenta con la posibilidad de actualizar la información de los resultados, así como borrarlos.

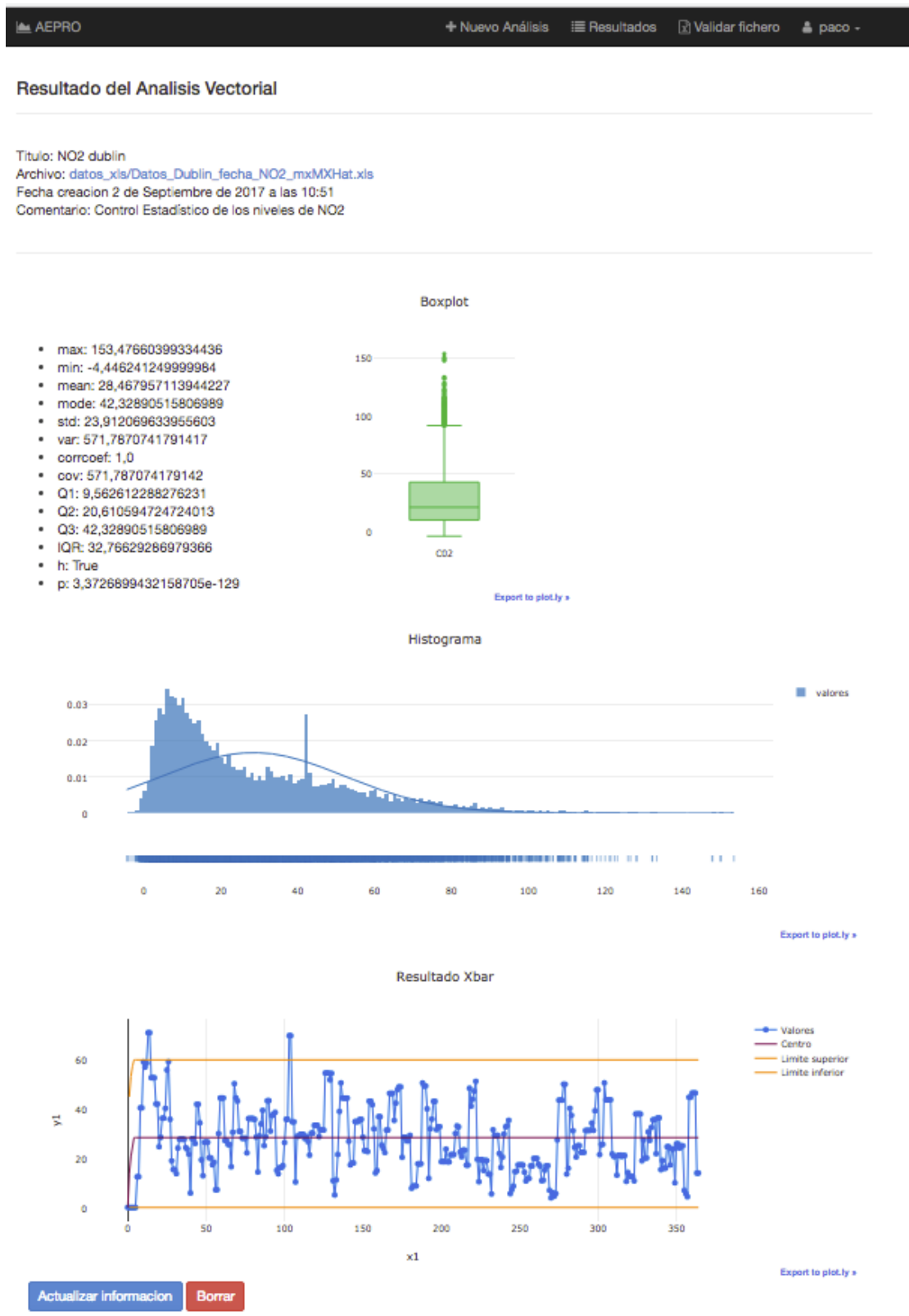
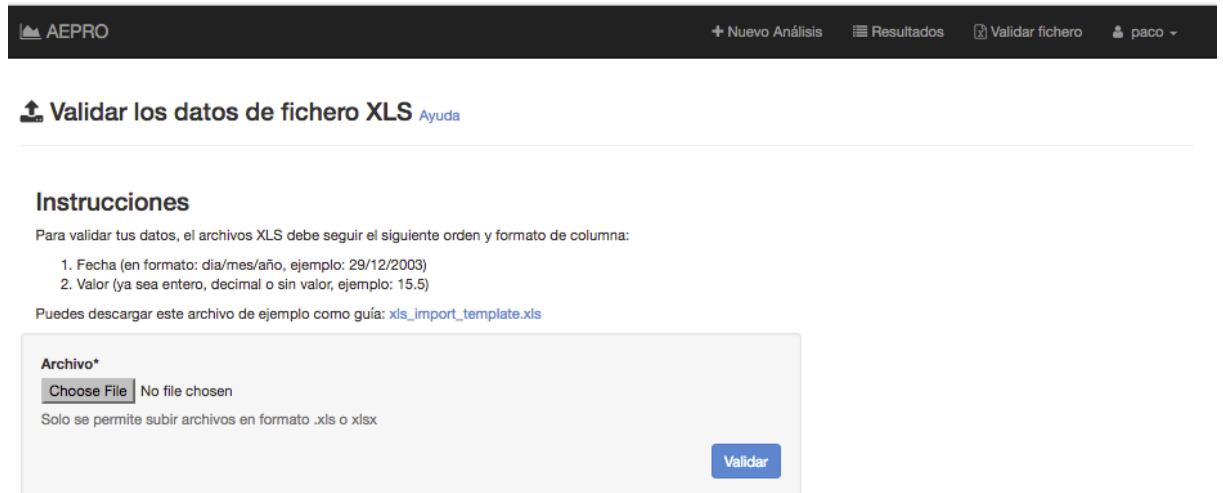


Figura 65. Resultado de un análisis CEP.

## Validar fichero

En esta apartado el usuario puede validar si el fichero de registro que quiere usar para analizar tiene un formato adecuado para la aplicación. Simplemente tiene que seleccionar el fichero y pulsar el botón validar, Figura 66. En caso de ser correcto, se muestra el mensaje de la Figura 67, y en caso de ser incorrecto el de Figura 68.



AEPRO + Nuevo Análisis Resultados Validar fichero paco

Validar los datos de fichero XLS [Ayuda](#)

### Instrucciones

Para validar tus datos, el archivos XLS debe seguir el siguiente orden y formato de columna:

1. Fecha (en formato: día/mes/año, ejemplo: 29/12/2003)
2. Valor (ya sea entero, decimal o sin valor, ejemplo: 15.5)

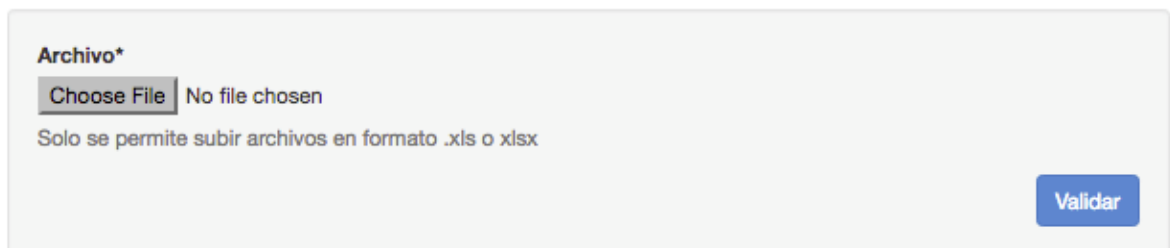
Puedes descargar este archivo de ejemplo como guía: [xls\\_import\\_template.xls](#)

**Archivo\***

No file chosen

Solo se permite subir archivos en formato .xls o xlsx

Figura 66. Sección para validar el formato del fichero.



**Archivo\***

No file chosen

Solo se permite subir archivos en formato .xls o xlsx

## ✓ Formato correcto

Figura 67. Respuesta para un formato correcto.



**Archivo\***

No file chosen

**Error: algun registro no esta en formato fecha**

Solo se permite subir archivos en formato .xls o xlsx

Figura 68. Respuesta para un formato incorrecto.

## Perfil de usuario

En esta última sección de usuario, (Figura 69), tiene acceso a la información de su perfil (Figura 70), puede modificar su contraseña (Figura 71), ponerse en contacto con los administradores de la aplicación (Figura 72) o cerrar la sesión actual.



Figura 69. Menú de acceso al perfil de usuario.

The page header shows 'AEPRO' on the left and navigation items '+ Nuevo Análisis', 'Resultados', 'Validar fichero', and 'paco' on the right. The main heading is 'Perfil'. Below it are four input fields: 'Nombre de usuario\*' with 'paco' and a lock icon; 'Nombre' with 'Francisco'; 'Apellidos' with 'Perez'; and 'Dirección de correo electrónico' with 'myemail@example.com'. At the bottom are 'Guardar' and 'Cancelar' buttons.

**Nombre de usuario\***

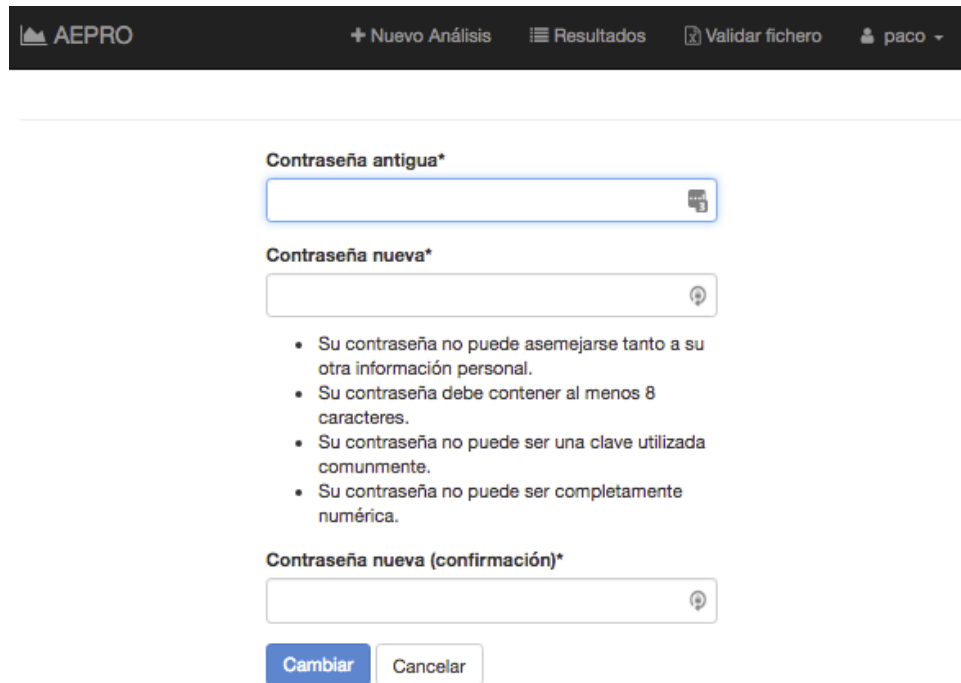
  
Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only.

**Nombre**

**Apellidos**

**Dirección de correo electrónico**

Figura 70. Información del perfil de usuario.



The screenshot shows the top navigation bar of the AEPRO application with the logo on the left and menu items: '+ Nuevo Análisis', 'Resultados', 'Validar fichero', and a user profile 'paco'. Below the navigation bar is a form for changing the password. It consists of three input fields: 'Contraseña antigua\*', 'Contraseña nueva\*', and 'Contraseña nueva (confirmación)\*'. Each field has a toggle icon on the right. Below the 'Contraseña nueva\*' field is a list of four requirements for the new password. At the bottom of the form are two buttons: 'Cambiar' (highlighted in blue) and 'Cancelar'.

AEPRO + Nuevo Análisis Resultados Validar fichero paco

Contraseña antigua\*

Contraseña nueva\*

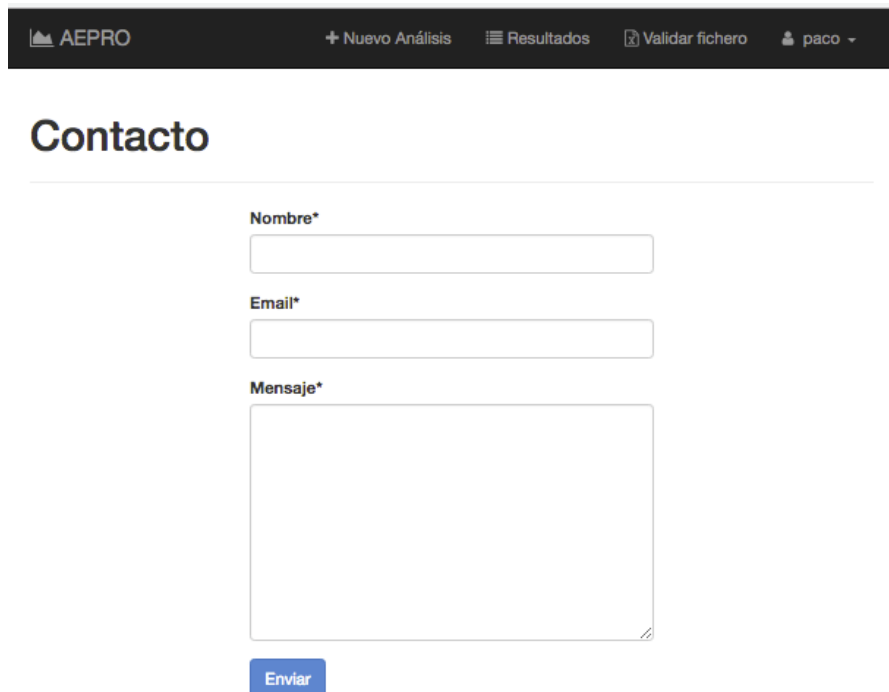
- Su contraseña no puede asemejarse tanto a su otra información personal.
- Su contraseña debe contener al menos 8 caracteres.
- Su contraseña no puede ser una clave utilizada comunmente.
- Su contraseña no puede ser completamente numérica.

Contraseña nueva (confirmación)\*

Cambiar Cancelar

Figura 71. Menú para el cambio de contraseña.

El sistema también cuenta con un formulario de contacto que permite enviar un correo al administrador/es para comentar cualquier duda que pueda surgir



The screenshot shows the top navigation bar of the AEPRO application. Below it is a section titled 'Contacto'. The form contains three input fields: 'Nombre\*', 'Email\*', and 'Mensaje\*'. The 'Mensaje\*' field is a larger text area. At the bottom of the form is a blue 'Enviar' button.

AEPRO + Nuevo Análisis Resultados Validar fichero paco

## Contacto

Nombre\*

Email\*

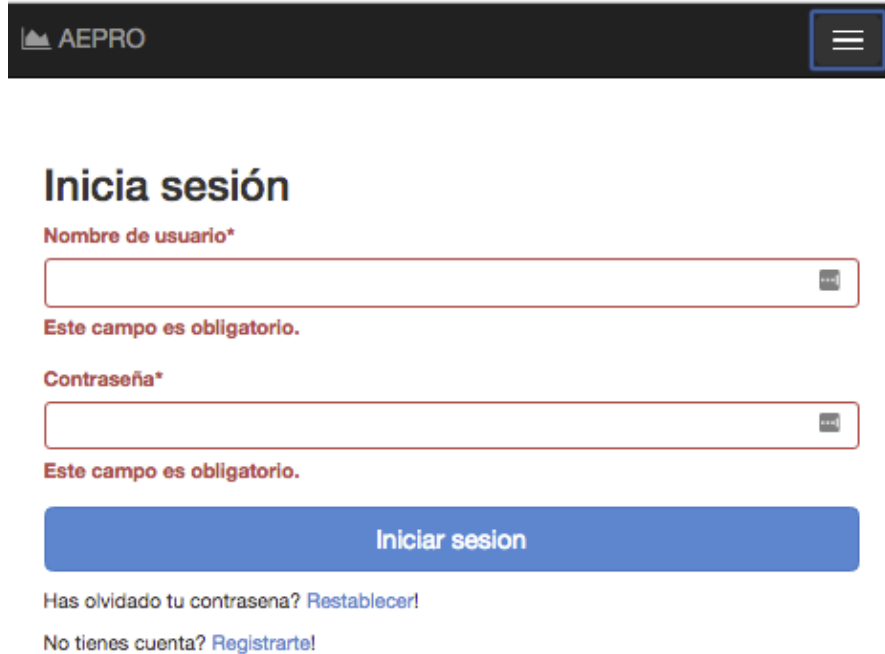
Mensaje\*

Enviar

Figura 72. Formulario de contacto.

## Restablecer contraseña

En el supuesto de necesitar recuperar la contraseña porque se ha olvidado el sistema permite restablecerla (Figura 73). Para ello solo será necesario indicar el email asociado a la cuenta (Figura 74) y se en enviará un correo con las instrucciones para restablecerla (Figura 75).



The screenshot shows the top navigation bar with the 'AEPRO' logo and a menu icon. Below it, the heading 'Inicia sesión' is displayed. There are two input fields: 'Nombre de usuario\*' and 'Contraseña\*'. Both fields have a red border and a red error message below them: 'Este campo es obligatorio.' A blue 'Iniciar sesión' button is positioned below the password field. At the bottom, there are two links: 'Has olvidado tu contraseña? [Restablecer!](#)' and 'No tienes cuenta? [Regístrate!](#)'.

Figura 73. Error de login en caso de no recordar la contraseña.



The screenshot shows the top navigation bar with the 'AEPRO' logo and a search icon. Below it, there are two input fields: 'Usuario' and 'Contraseña', followed by 'Entrar' and 'Regístrate' buttons. The main content area features a label 'Correo electrónico\*' above a text input field. Below the input field is an 'Enviar' button.

Figura 74. Email proporcionado por el usuario para restablecer contraseña.



The screenshot shows the top navigation bar with the 'AEPRO' logo and a search icon. Below it, there are two input fields: 'Usuario' and 'Contraseña', followed by 'Entrar' and 'Regístrate' buttons. The main content area displays a message: 'Un email con instrucciones para restablecer tu contraseña ha sido enviado.'

Figura 75. Mensaje de envío de instrucciones para restablecimiento de contraseña.

## Anexo II. Prototipos de la interfaz de usuario

Este segundo anexo se ha usado para completar la sección 5.5 Implementación de la interfaz de usuario.

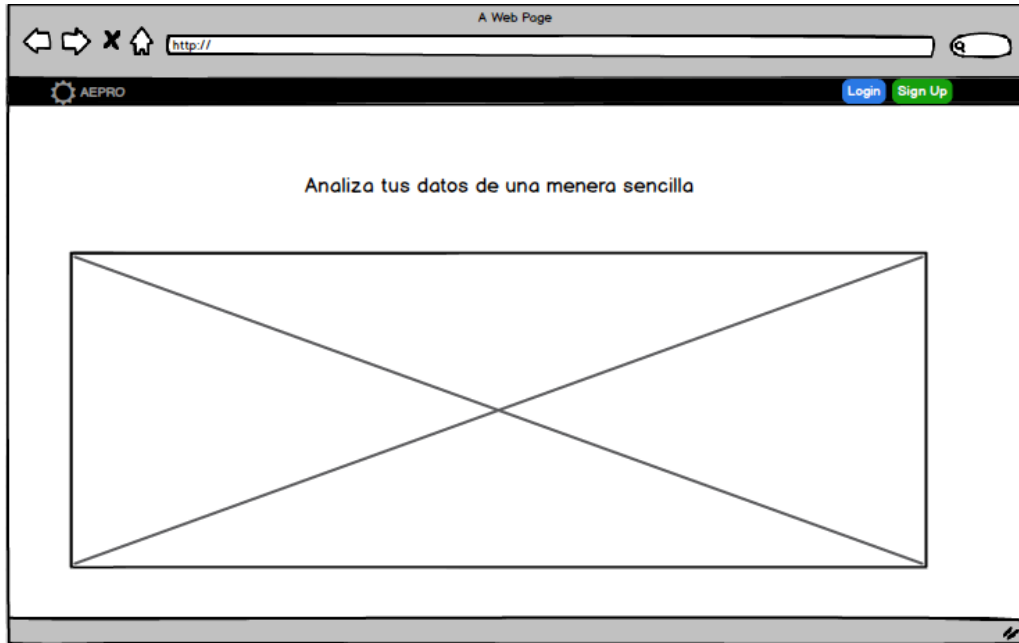


Figura 76. Mockup: Landig page

Pantalla	Página de Inicio
Descripción	La página de inicio o <i>landing page</i> es la primera página que verá un usuario al acceder a la aplicación. Desde ella el usuario podrá acceder a la aplicación o darse de alta como nuevo usuario
RQF asociados	RQF – 13 RQF – 15
RQNF asociados	

Tabla 49. Descripción de la página de inicio

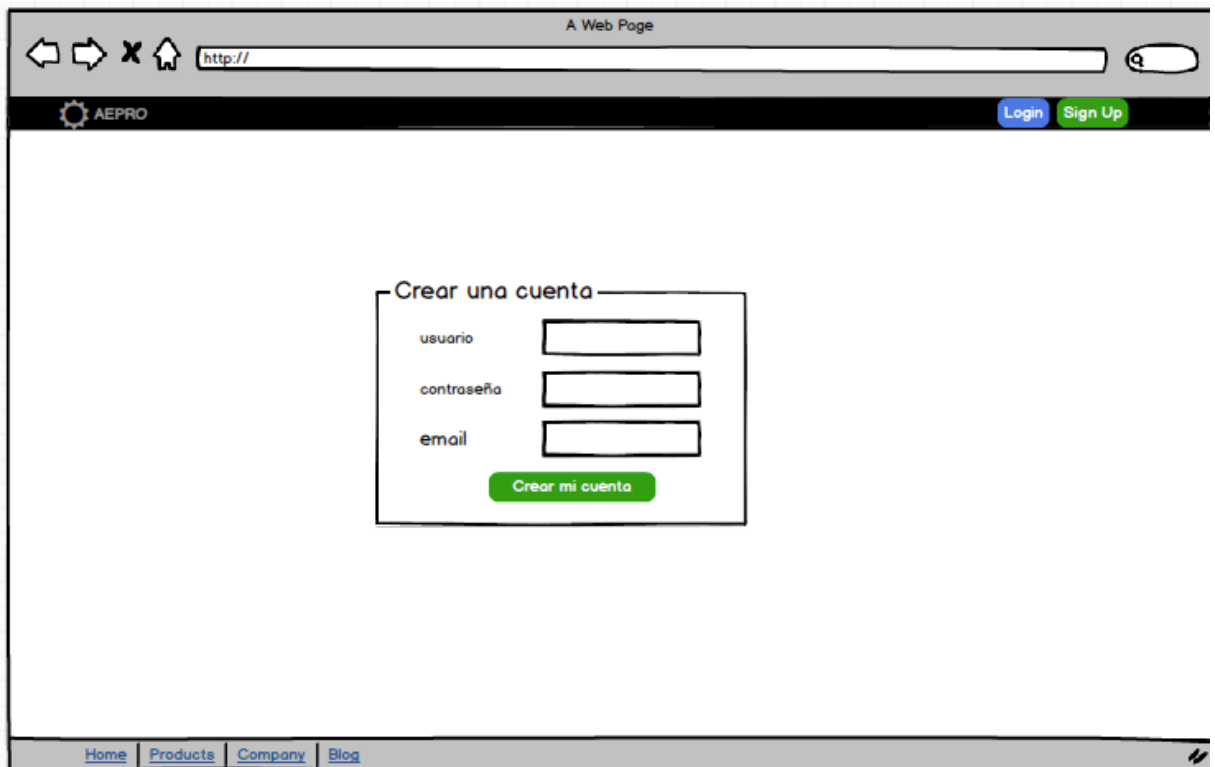


Figura 77. Mockup: página de registro en la aplicación.

Pantalla	Página de alta de nuevos usuarios
Descripción	Desde esta página un usuario podrá darse de alta únicamente usando un nombre de usuario y correo electrónico válido.
RQF asociados	RQF-17
RQNF asociados	

Tabla 50. Descripción de la página de alta de nuevos usuarios.

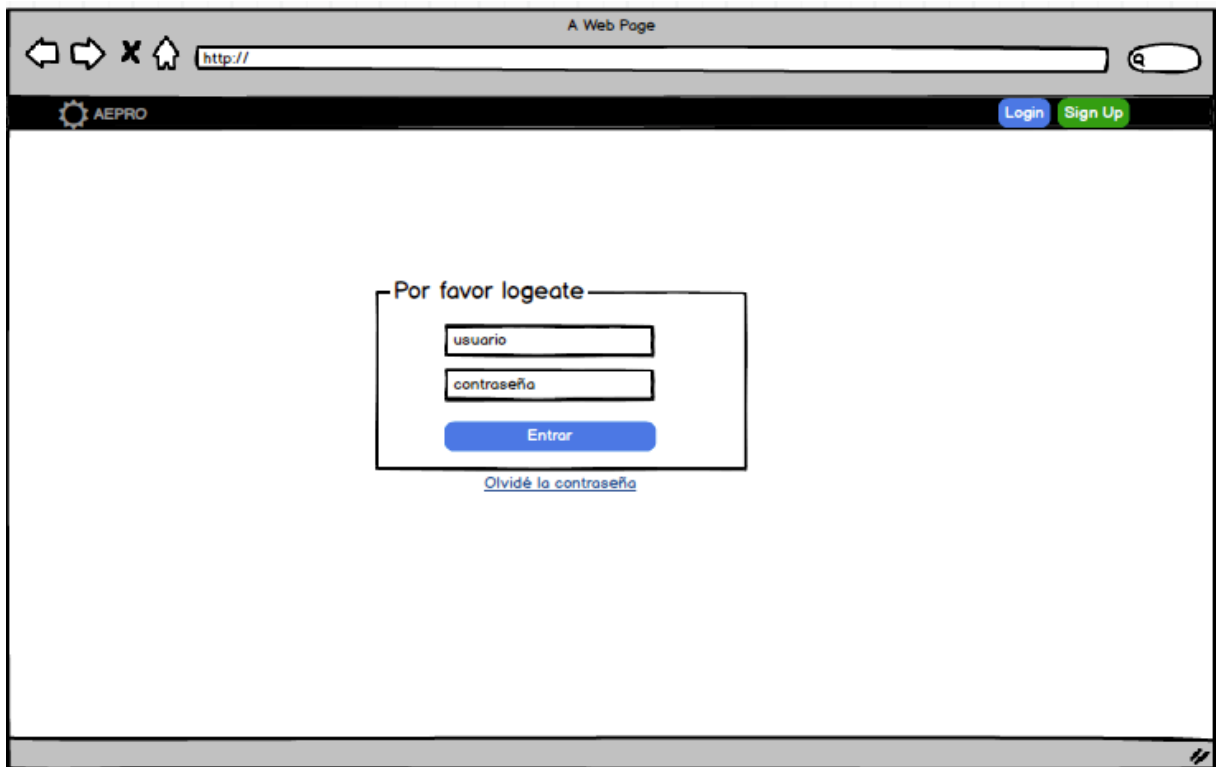


Figura 78. Mockup: página de login en la aplicación.

Pantalla	Página de login
Descripción	Desde esta página un usuario que tenga una cuenta ya activada, podrá acceder al contenido de la aplicación.
RQF asociados	RQF 17 RQF 19 RQF 20 RQF 21
RQNF asociados	

Tabla 51. Descripción de la página de login.

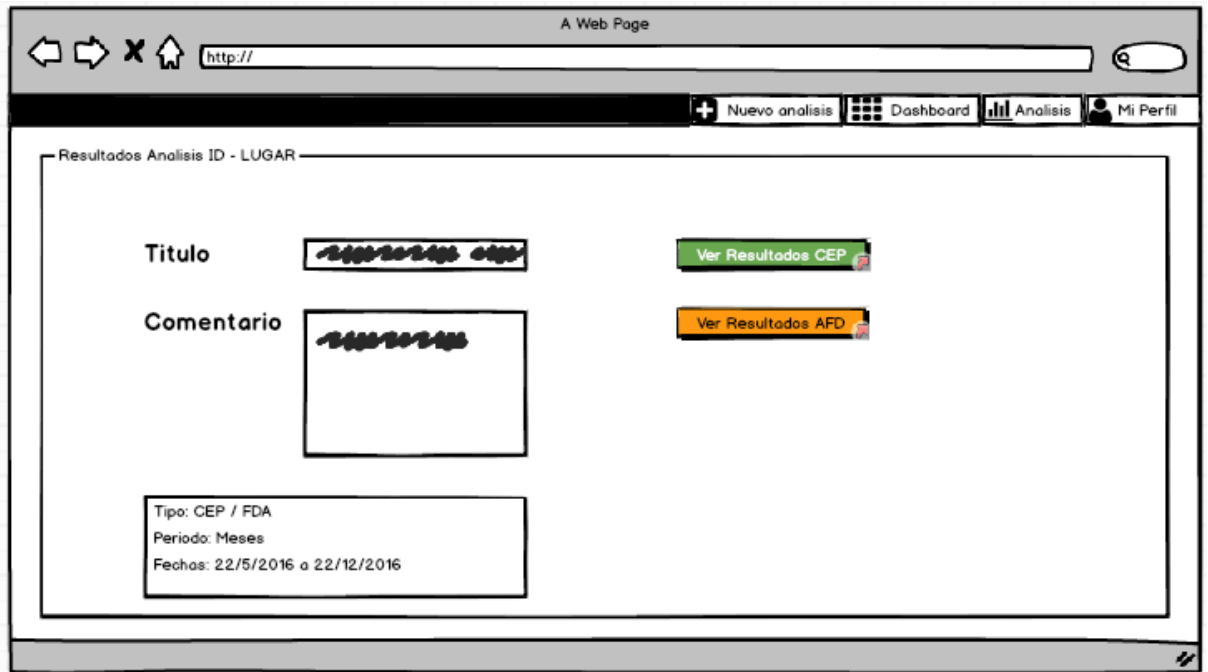


Figura 79. Mockup Detalle de un análisis

Pantalla	Resultado de un análisis
<b>Descripción</b>	Dentro de la descripción del propio análisis, se puede ver título y comentario: campos que pueden modificarse. Y la información relativa a análisis como tipo de análisis seleccionado periodo y fechas a analizar. En la parte derecha aparecen 1 o 2 botones en función del tipo/s de análisis seleccionadas CEP y/o AFD. Mediante colores se indica si los resultados de los análisis están disponibles (verde) o están en procesos de cálculo (naranja)
<b>RQF asociados</b>	RQF - 12
<b>RQNF asociados</b>	

Tabla 52. Descripción de la información de asociada a un análisis.

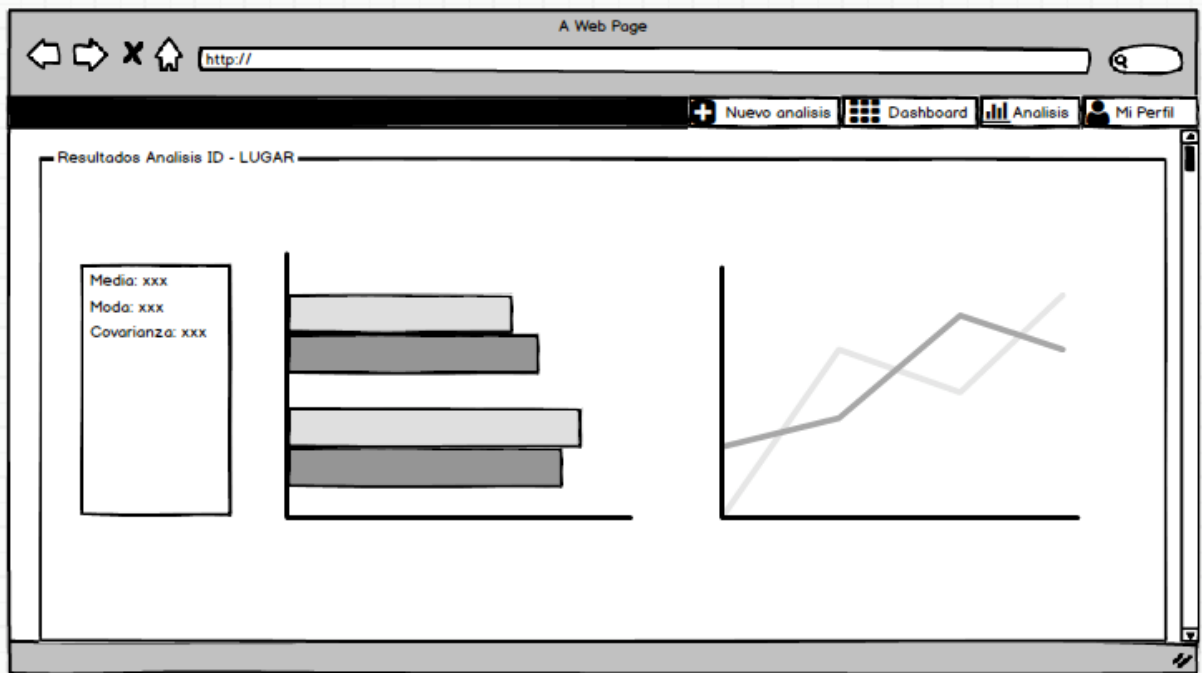


Figura 80. Mockup resultado análisis Control Estadístico de procesos

Pantalla	Información resultante del análisis: CEP
Descripción	Esta apartado permite mostrar los gráficos e información resultante de proceso de análisis CEP. Los gráficos permitirán cierta interacción, que ayude al usuario a visualizarlos.
RQF asociados	RQF-12 RQF-14 RQF-16
RQNF asociados	

Tabla 53. Descripción de la información resultante del tipo de análisis CEP.

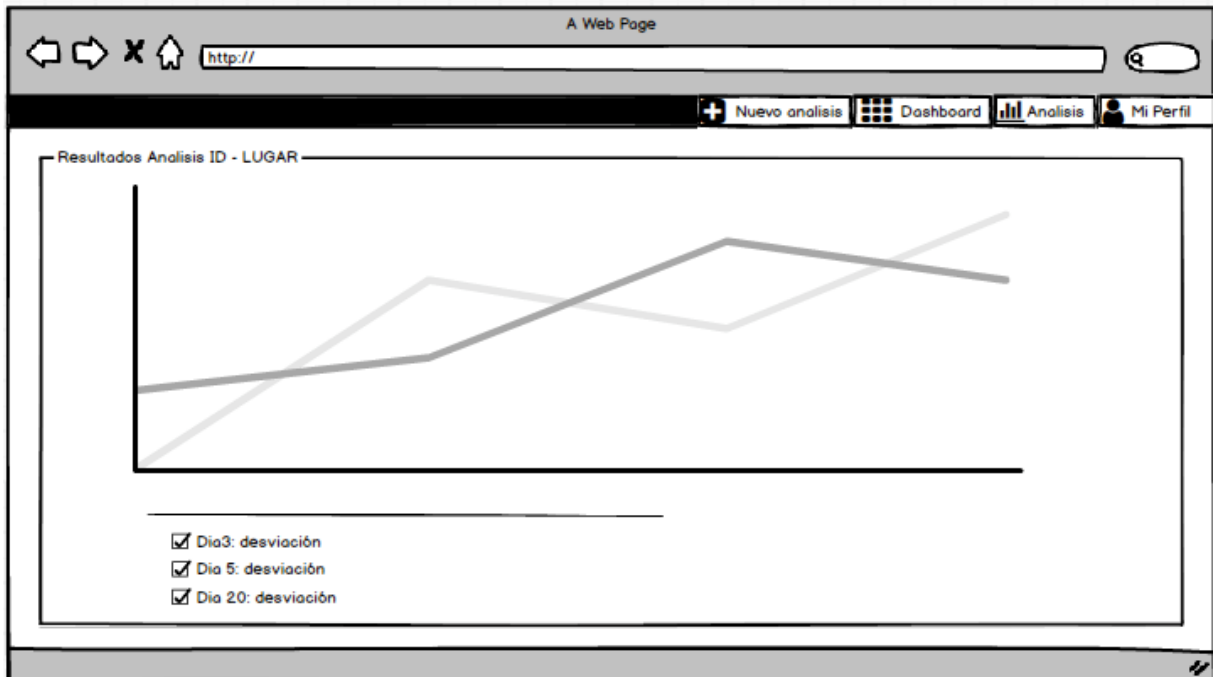


Figura 81. Mockup resultado análisis: Análisis Funcional de Datos.

Pantalla	Información resultante del análisis: FDA
Descripción	Esta apartado permite mostrar los gráficos e información resultante de proceso de análisis FDA. Los gráficos permitirán cierta interacción, que ayude al usuario a visualizarlos.
RQF asociados	RQF-12 RQF-15 RQF-16
RQNF asociados	

Tabla 54. Descripción de la información resultante del tipo de análisis FDA.

## Anexo III. Pruebas de subsistema

### Subsistema alta de nuevos usuarios

Desde el apartado de registro un usuario anónimo puede darse de alta en el sistema, Figura 82, rellenando el formulario.

**Registrate:**

Nombre de usuario\*

alberto

Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only.

E-mail\*

alsanche@unizar.es

Contraseña\*

.....

Contraseña (confirmación)\*

.....

Para verificar, introduzca la misma contraseña anterior.

Enviar

Figura 82. Página de registro.

En la bandeja de entrada de su correo el usuario recibe un correo como el de la Figura 83. Como no se cuenta con un dominio, se hace uso de uno indicado por defecto (example.com) para no tener problemas.



Figura 83. Email recibido para la activación de la cuenta.

Una vez accedido al enlace, ya está la cuenta activa y puede verse en la parte superior como el usuario se encuentra dentro de la aplicación (Figura 84).



Figura 84. Mensaje de activación de la cuenta.

### Subsistema acceso a la aplicación

En la parte superior izquierda de la aplicación, se han especificado dos campos, usuario y contraseña, donde el usuario debe rellenar y pulsar el botón de entrar, Figura 85.



Figura 85. Landing page de acceso a la aplicación.

En caso de intentar acceder con unas credenciales erróneas o intentar acceder mediante las urls directamente al contenido, el sistema redirige al *login* que le permita acceder, como muestra la Figura 86.

Figura 86. Página de login en caso de producirse algún error.

Una vez que se introducen las credenciales correctas, se tiene acceso a la aplicación. Figura 87.

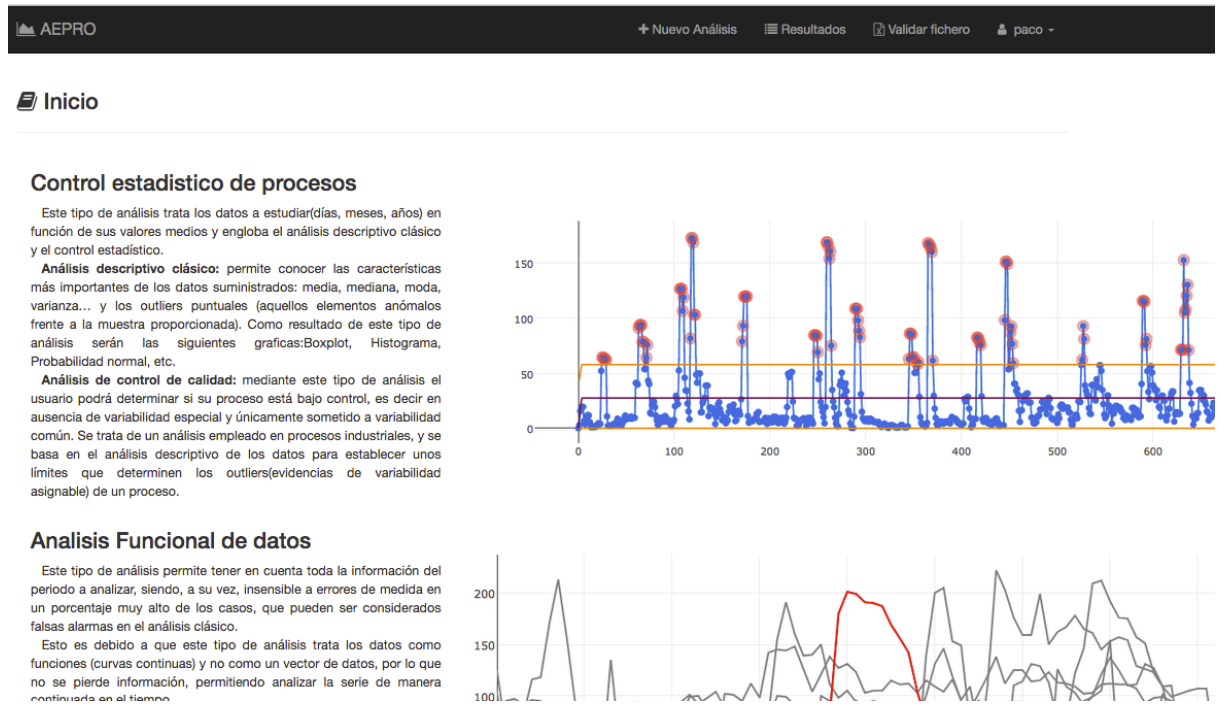


Figura 87. Página de inicio para usuarios autenticados.

### Subsistema gestión del perfil de usuario

El usuario puede acceder a la información de su perfil, en la parte superior izquierda de la aplicación, véase la siguiente Figura 88.

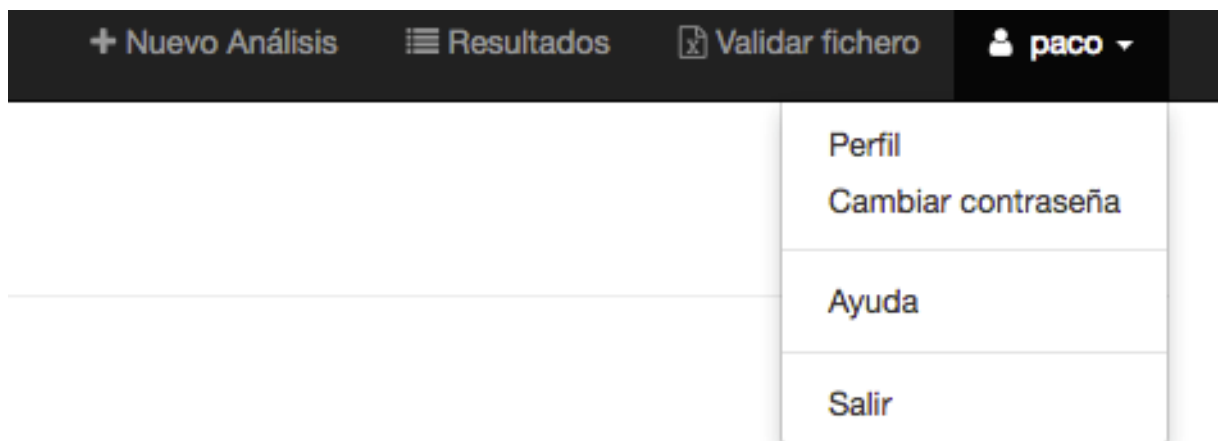


Figura 88. Menú desplegable para el perfil de usuarios.

El sistema muestra la información proporcionada por el usuario en el proceso de alta de nuevo usuario.

## Perfil

---

**Nombre de usuario\*** 

Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only.

**Nombre****Apellidos****Dirección de correo electrónico**

Figura 89. Información del usuario.

### Subsistema creación de un nuevo análisis

Para la creación de un nuevo análisis, Figura 90, solamente es necesario rellenar el formulario con la información solicitada.

The screenshot shows the 'Crear Análisis' form in the AEPRO system. The form is titled '+ Crear Análisis' and contains the following fields and options:

- Título:** NO2 dublin
- Selecciona Periodo:** Mes
- Tipo analisis:**
  - Control Estadístico de Procesos
  - Analisis Funcional de Datos
- Fecha inicio:** 2013-01-01 00:00
- Fecha paso:** 2013-01-01 00:00
- Fecha fin:** 2013-12-31 00:00
- Indica un comentario si es necesario:** Escribe aquí tu comentario
- File:** Choose File No file chosen. Solo se permite subir archivos en formato XLS!!!!

At the bottom of the form are two buttons: 'Calcular' and 'Reset'. A calendar widget is open, showing the month of December 2013. The date 31 is selected.

Figura 90. Formulario para la creación de un nuevo análisis.

Si todos los datos son correctos el sistema muestra el mensaje de la Figura 91.



"Análisis creado correctamente."

Figura 91. Mensaje de análisis creado correctamente.

### Subsistema lista de resultados

En la lista de resultados, Figura 92, es posible saber el estado en el que se encuentran los distintos análisis creados.



Figura 92. Lista de resultados.

Si se encuentran disponibles los resultados se puede acceder a la información: Figura 93  
Figura 94

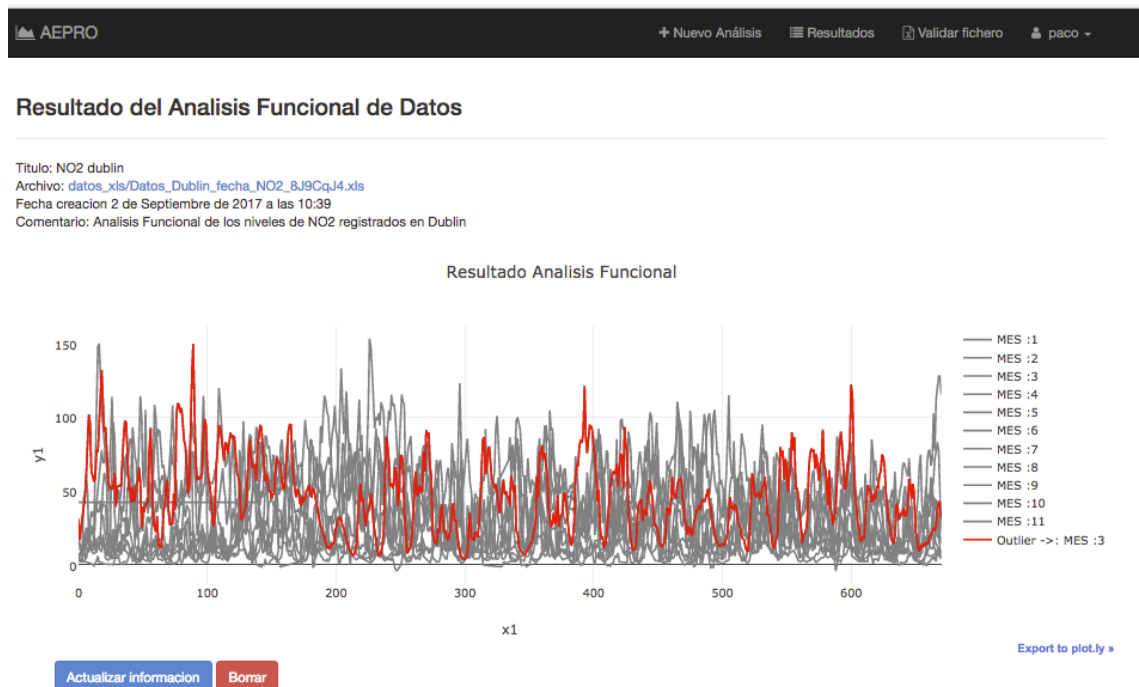


Figura 93. Gráficos resultantes para un análisis FDA.

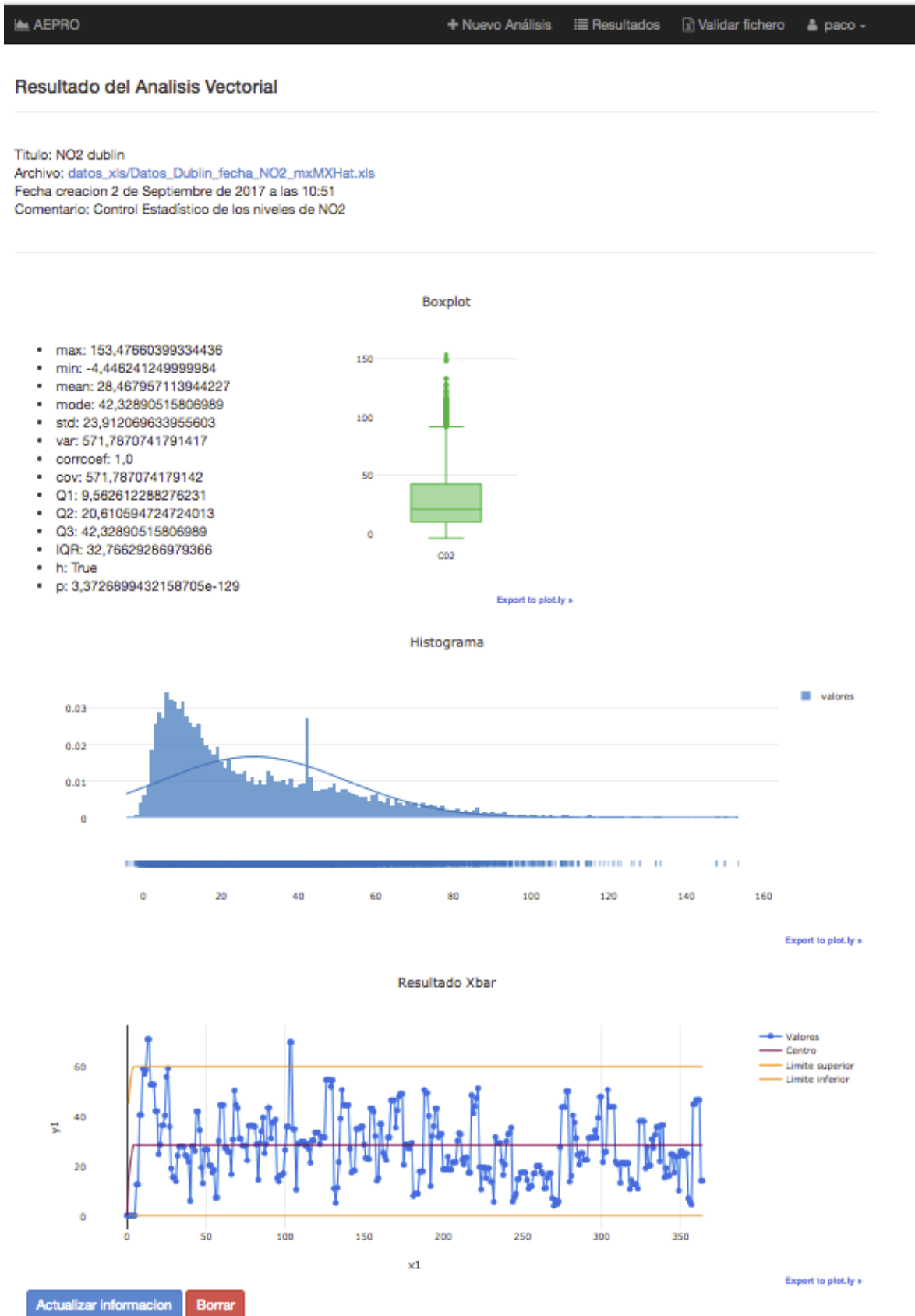


Figura 94. Gráficos resultantes para un análisis CEP.

## Subsistema validación de ficheros

Este apartado de la aplicación permite validar el formato del fichero. Para ello el usuario selecciona un fichero, Figura 95, y el sistema lo procesa.

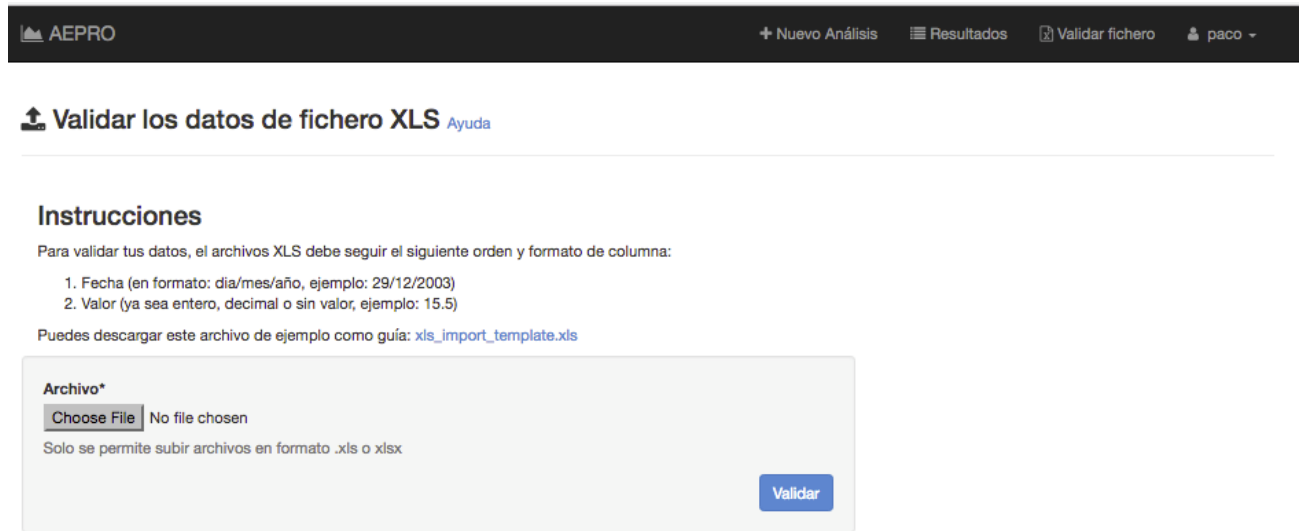


Figura 95. Apartado de validación de formato de ficheros.

Si cumple con los requisitos necesarios, el sistema muestra el mensaje de la Figura 96.

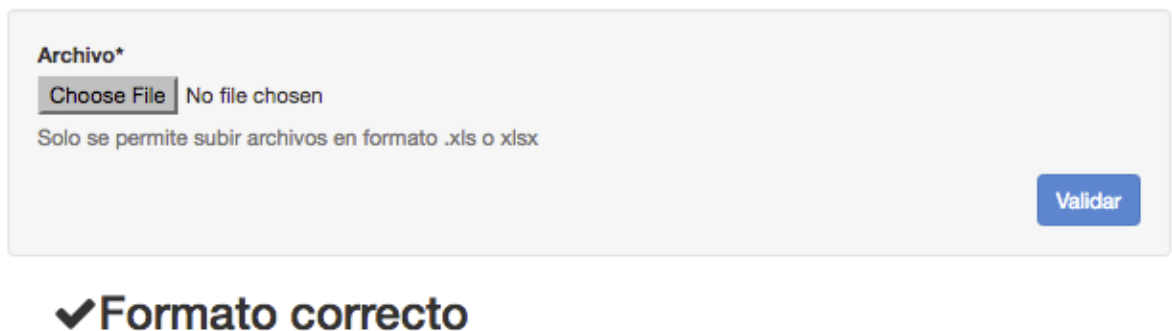


Figura 96. Mensaje resultante de la validación del formato de un fichero.