



TRABAJO FIN DE GRADO
GRADO EN INGENIERÍA EN INFORMÁTICA Y
ADMINISTRACIÓN Y DIRECCIÓN DE EMPRESAS

Visualización de datos de los Objetivos de Desarrollo Sostenible

Autor

Santiago Carbó García

Director

Carlos Ureña Almagro



Escuela Técnica Superior de Ingenierías Informática y de
Telecomunicación

Granada, Noviembre de 2022



Visualización de datos de los Objetivos de Desarrollo Sostenible

Autor

Santiago Carbó García

Director

Carlos Ureña Almagro

Visualización de datos de los Objetivos de Desarrollo Sostenible

Realizado por Santiago Carbó García

Palabras clave: ODS, API Canvas, JavaScript, WebGL, HTML5, CSS

Resumen

La representación de los datos es una rama de tamaño considerable dentro del desarrollo de sistema gráficos. Asimismo, el Desarrollo Sostenible es un tópico de suma relevancia en la época contemporánea y, por ende, la representación de sus datos. Es por ello por lo que este documento tratará de analizar y explicar el funcionamiento de una aplicación que tiene el objetivo de mostrar dichos datos utilizando especificación de suma utilidad como es WebGL y, dentro de ella, la herramienta API Canvas. La aplicación consiste en un programa JavaScript que utiliza dicha herramienta para la representación de gráficos de dos dimensiones. Para ello, los datos se recogen de un fichero obtenido de la base de datos del Instituto Nacional de Estadística, con un formato en particular. Después son procesados y almacenados para su posterior visualización. Asimismo, cada gráfica dispone de unos parámetros de estilo que pueden ser configurados *a posteriori*. Se han desarrollado tres tipos de gráficos para la visualización de dichos datos: *gráficos de barras*, *gráficos de líneas* y *gráficos circulares*, los cuales permiten tres formas muy intuitivas de reproducir la información. Para la elaboración de los ejes de coordenadas en los dos primeros tipos de gráficos se tendrán en cuenta algunas consideraciones. Primero, el cálculo del número de líneas horizontales necesarias para representar los valores referencia del eje vertical y, segundo, el número de valores mostrados en la gráfica teniendo en cuenta que puede no haber espacio para visualizarlos todos. Es por ello por lo que se usan algunos parámetros como el valor máximo de la serie o el número total de valores.

Data visualization of the Sustainable Development Goals

Written by Santiago Carbó García

Keywords: ODS, API Canvas, JavaScript, WebGL, HTML5, CSS

Abstract

Data representation is a branch of considerable size within the development of graphic systems. Likewise, Sustainable Development is a topic of great relevance in contemporary times and, therefore, the representation of its data. That is why this document will try to analyze and explain the functioning of an application that has the goal of displaying such data using a very useful specification such as WebGL and, within it, the Canvas API tool. The application consists of a JavaScript program that uses this tool in order to represent two-dimensional charts. For the purpose of doing this, the data are collected from a file, obtained from the database of the Spanish National Institute of Statistics, with a particular format. Then the data are processed and stored for later viewing. Likewise, each chart has style parameters that can be configured. Three types of charts have been developed for the visualization of such data: bar charts, line charts and pie charts, which represent three intuitive ways of reproducing information. For the elaboration of the coordinate axes in the first two types of charts, some considerations will be taken into account. First, the calculation of the number of horizontal lines necessary to represent the reference values of the vertical axis and, secondly, the number of values shown in the chart, taking into account that there may not be space to display them all. That is why some parameters are used, such as the maximum value of the series or the total number of values.

Yo, **Santiago Carbó García**, alumno de la titulación *Doble Grado en Ingeniería Informática y Administración y Dirección de Empresas* de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 77561011C, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Santiago Carbó García

Granada, a 1 de septiembre de 2022

D. Carlos Ureña Almagro, Profesor del Área de [] de la Universidad de Granada.

Informa:

Que el presente trabajo, titulado ***Visualización de datos de los Objetivos de Desarrollo Sostenible***, ha sido realizado bajo su supervisión por **Santiago Carbó García**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a 1 de septiembre de 2022.

El director: Dr. Carlos Ureña Almagro

Carlos Ureña Almagro

Agradecimientos

A mi madre y a mi hermana por apoyarme desde que nací, a mis abuelos maternos por cuidarme desde que era pequeño, a mi abuelo Pepe por venir siempre a visitarme desde tan lejos y a María por cuidar de mí todos los días. Sin ellos mi vida sería mucho más difícil.

Índice general

1. INTRODUCCIÓN	1
1.1. Motivación	1
1.2. Objetivos	1
1.2.1. Objetivos generales	1
1.2.2. Objetivos específicos	1
2. PLANIFICACIÓN	1
2.1. Metodología Scrum	
2.2.	
3. ESPECIFICACIÓN DE LOS REQUERIMIENTOS	1
3.1. Requisitos funcionales	1
3.2. Requisitos no funcionales	1
3.3. Bocetos iniciales	1
4. DISEÑO	
4.1. Diagrama de clases	1
4.2. Diagramas de interacción	1
4.3. Bocetos de las interfaces de usuario	1
4.4. Estructuras de datos fundamentales	1
4.5. Desarrollo de algoritmos no triviales	1
5. IMPLEMENTACIÓN	
5.1. Descripción de las clases	1
5.2. Descripción de los métodos de las clases	1
6. CONCLUSIONES Y VÍAS FUTURAS	1
6.1. Evaluación del cumplimiento de los objetivos	1
6.1.1. Evaluación del cumplimiento de los objetivos generales	1
6.1.2. Evaluación del cumplimiento de los objetivos específicos	1
6.2. Vías futuras de desarrollo	1

1. Introducción

1.1. Motivación

El desarrollo sostenible es de gran relevancia hoy en día, afectando, entre otras cosas, al derecho

y a la informática. Es importante tener en cuenta el consumo energético a la hora de la construcción de sistemas y redes, un consumo que deberemos ir reduciendo con el transcurso de los años para adaptarnos, no solo a la legislación, sino a las necesidades energéticas.

Estas gráficas también deben estar respaldadas por fuentes fiables de información, ya que gran parte de esta no proviene de fuentes que garanticen que la información es veraz. Es por ello por lo que se deben utilizar siempre datos de fuentes fiables (del Instituto Nacional de Estadística o Naciones Unidas, por ejemplo). Su influencia y su impacto son lo suficientemente importantes como para preocuparse por la fiabilidad de los datos. Es por todo ello que un desglose de estos datos y su respectiva representación por el software pertinente es de suma importancia. La estructura de los gráficos, los colores utilizados para ello, el espaciado, etc., serán buenos indicadores de una correcta representación.

Los datos por sí solos carecen de significado. Estos deben ser visualizados e interpretados correctamente mediante el software necesario. Este software no solo debe interpretar y visualizar datos, sino que debe hacerlo correctamente, es decir, debe moldearse a las necesidades cambiantes del usuario, de ahí la importancia de elaborar interfaces correctas en términos de usabilidad y accesibilidad. Por ende, una parte esencial de la visualización de los datos es la comunicación, ya que una imagen tiene el potencial de comunicar de una forma más efectiva que la palabra. Actualmente, debido a la rapidez con la que se desarrolla información y la capacidad de acceso por parte del usuario a este, el consumo de información se ha disparado de manera exponencial. Debido a ello, estamos expuestos diariamente a grandes cantidades de datos, los cuales tienen formatos y estilos diferentes para su representación, lo cual hace aún más difícil su identificación.

Finalmente, aclarar la importancia de las hojas de cálculo para la obtención de información de una forma clara y seguro. Con el objetivo de obtener datos para la representación de gráficos estas son de gran ayuda para almacenarlos. Muchos bancos de información utilizan hojas de cálculos para exportar información con respecto a gráficos, hecho por el cual me he guiado para el desarrollo de la aplicación. Permiten exportar dichas hojas de cálculos de distintas maneras para su procesamiento (celdas separadas por punto y coma, por tabuladores, etc.). Para interpretar esta información, los gráficos de barras, líneas y circulares conforman una de las formas más simples e intuitivas para adquirir esta información.

1.2. Objetivos

Al inicio de todo proyecto es fundamental establecer una serie de objetivos generales y específicos. Es importante poner el enfoque en los aspectos más importantes y cuantificar la relevancia de cada idea. Unos objetivos fiables nos permitirán, en el futuro, el cumplimiento de estos y estimar si son necesarios algunos cambios.

1.2.1. Objetivos generales

A modo de síntesis, una definición correcta de los objetivos de este proyecto sería “desarrollar una aplicación interactiva que permite visualizar a los usuarios gráficos que representan información relacionada con los Objetivos del Desarrollo Sostenible y sus diferentes indicadores”. No obstante, a modo de desglose, sería posible identificar algunos objetivos generales:

- *Desarrollo de una aplicación web intuitiva e interactiva.* La aplicación realizada contiene una interfaz de usuario intuitiva que permitirá que el usuario pueda utilizarla con facilidad.
- *Visualización de gráficos en dos dimensiones.* Uno de los fines fundamentales del proyecto consiste en el visualizado de gráficos de dos dimensiones. Estos podrán ser gráficos de barras, de líneas o circulares
- *Procesamiento de datos almacenados en una hoja de cálculo.* Los datos utilizados para la representación de gráficos serán previamente obtenidos y procesados de una hoja de cálculo.
- *Visualización de información auxiliar.* Información acerca del uso de la aplicación o la procedencia de los ficheros utilizados. También será posible acceder a información relacionada con el proyecto.

1.2.2. Objetivos específicos

Una vez explicitados los objetivos generales es posible obtener una visión general del alcance del proyecto. Esto nos permitirá tener una visión clara de la magnitud de este. No obstante, los objetivos específicos nos permiten desmenuzar los anteriores en unidades más pequeñas y fáciles de alcanzar, con el objetivo de trazar metas a corto plazo.

Los objetivos específicos del proyecto serían los siguientes:

- *Elaboración y visualización de cada uno los elementos de las gráficas.* Cada elemento de las gráficas deberá ser elaborado por el programa, incluyendo los ejes de coordenadas en los gráficos de barras y líneas y el texto que representa valores y etiquetas.
- *Elaboración de un algoritmo que permita optimizar la visualización de elementos en el gráfico.* Es posible que se deban representar una gran cantidad de valores, de forma que es necesario optimizar la aparición de estos en el gráfico.
- *Desarrollo de botones para agregar y eliminar series de datos y gráficas.* Se podrá añadir y eliminar series de datos y gráficas mediante sus botones pertinentes.
- *Opción que permita la carga de ficheros mediante “drag and drop”.* Se podrán cargar ficheros por dos vías: o bien seleccionándolo en el explorador de archivos por la vía clásica o arrastrando el fichero al cuadro.
- *Desarrollo de opciones de estilo para cada gráfica.* Cada gráfica dispondrá de una serie opciones para poder cambiar parámetros de estilos tales como el grosor de línea o el gradiente.
- *Desarrollo de un efecto de tres dimensiones para los gráficos de barras y circulares.* Una de las opciones permitidas será activar el efecto de tres dimensiones, que permitirá darle profundidad a las barras y al gráfico circular.
- *Representación de las secciones mediante colores fáciles de combinar.* Los colores que representarán las barras en el gráfico de barras y las secciones en los gráficos circulares deberán ser colores pastel que combinen fácilmente con la gráfica.
- *Visualización de los criterios utilizados en la serie de datos.* Será posible conocer aquellos criterios utilizados para la elaboración de las series de datos.

2. Planificación

3. Especificación de requerimientos

3.1. Requisitos funcionales

Se presenta un listado de requisitos funcionales de la aplicación:

RF.1	Inicio de la aplicación web
Descripción	Posibilidad de ejecutar la aplicación web
Datos de entrada	Ninguno
Datos de salida	Ninguno

RF.2	Cierre de la aplicación web
Descripción	Posibilidad de terminar la aplicación web
Datos de entrada	Ninguno
Datos de salida	Ninguno

RF.3	Visualización de gráficos de barras
Descripción	Los datos importados del archivo .csv podrán visualizarse en un gráfico de barras
Datos de entrada	Datos procesados obtenidos del archivo .csv
Datos de salida	Gráfico de barras que representa los datos pertinentes

RF.4	Visualización de gráficos de líneas
Descripción	Los datos importados del archivo .csv podrán visualizarse en un gráfico de líneas
Datos de entrada	Datos procesados obtenidos del archivo .csv
Datos de salida	Gráfico de líneas que representa los datos pertinentes

RF.5	Visualización de gráficos circulares
Descripción	Los datos importados del archivo .csv podrán visualizarse en un gráfico circular
Datos de entrada	Datos procesados obtenidos del archivo .csv
Datos de salida	Gráfico circular que representa los datos pertinentes

RF.6	Conocer el número de variables utilizadas para cada gráfico
Descripción	
Datos de entrada	Datos procesados obtenidos del archivo .csv
Datos de salida	Número indicando el número de variables

RF.7	Insertar archivos “.csv”, con el objetivo de visualizar dichos datos
Descripción	Se podrán utilizar datos organizados en archivos .csv, con el fin de utilizarlos para las gráficas
Datos de entrada	Archivo .csv
Datos de salida	Datos procesados en su respectiva estructura de datos

RF.8	Conocer información relacionada con el proyecto en un apartado de información
Descripción	Apartado que permite conocer información relacionada con el proyecto
Datos de entrada	Ninguno
Datos de salida	Ninguno

3.2. Requisitos no funcionales

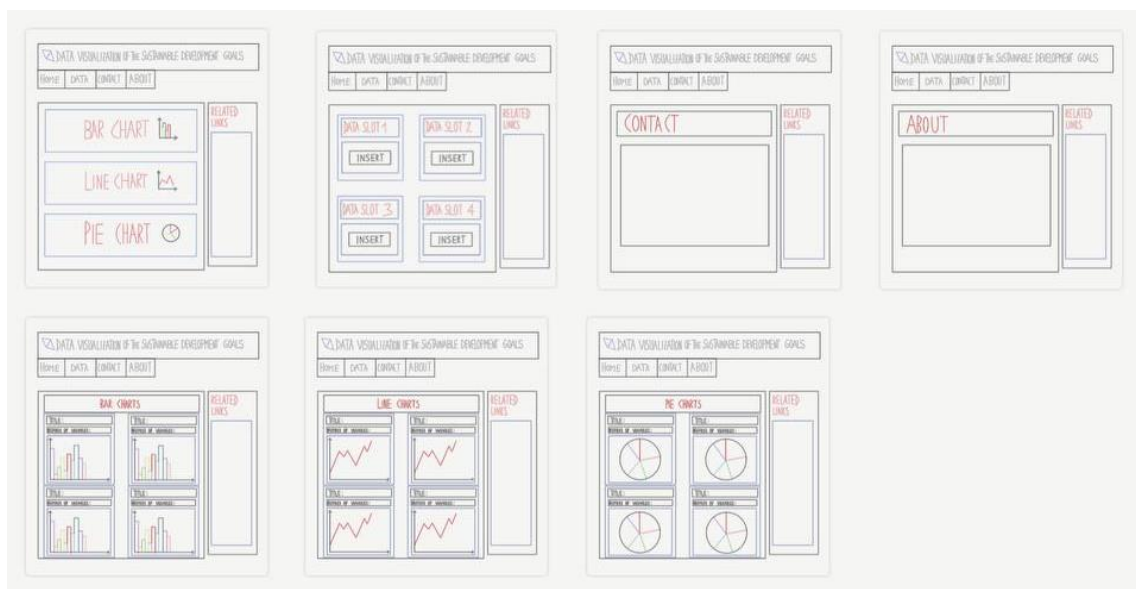
RNF.1	La aplicación se ejecutará en un navegador web actual en cualquier sistema operativo, mediante la visualización de una página web con los controles y gráficas
Descripción	El desarrollo de la aplicación web se realizará utilizando las herramientas web pertinentes con el fin de la visualización de gráficas

RNF.2	La aplicación tendrá fácil navegabilidad
Descripción	La aplicación será usable en términos de navegabilidad. Se podrá acceder a todo con pocos <i>clicks</i> .

RNF.3	La aplicación se encontrará disponible en inglés
Descripción	El lenguaje del que dispondrá la aplicación será el inglés

3.3. Bocetos iniciales

Los bocetos son una forma sencilla e intuitiva para un primer contacto con la aplicación. A continuación, se muestran los bocetos de la aplicación:



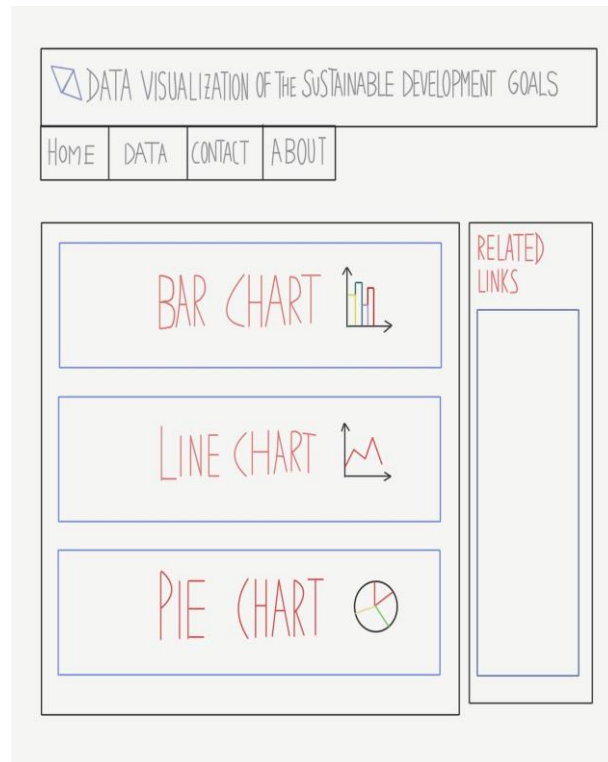
Con el objetivo de simplificar el boceto, se han sintetizado algunos conceptos de la versión final de la aplicación. Un ejemplo importante es el hecho de que se han dibujado únicamente cuatro gráficas. No obstante, la aplicación web será un sistema en el que será posible visualizar gráficas sin que exista un límite preestablecido en estas.

La aplicación web dispondrá de una pestaña “*Home*”, la cuál será la pestaña de inicio. En ella se podrá elegir el tipo de gráfico que se desea: gráfico de barras, de líneas o circular. Una vez se pinche en algunas de estas opciones, se redirigirá a la pestaña pertinente donde se mostrarán los gráficos con los datos establecidos (en caso de no haber introducido ningún dato, existirán unos datos preestablecidos).

También se dispone de una pestaña “*Data*”, la cual está desarrollada con la intención de indicar cuáles son los datos que queremos para cada una de las gráficas. Como se ha indicado previamente, no existe un límite preestablecido de gráficas, de forma que una vez se inserte un archivo de datos aparecerá una ranura que permitirá introducir otro archivo de datos, y así sucesivamente.

A modo de desglose:

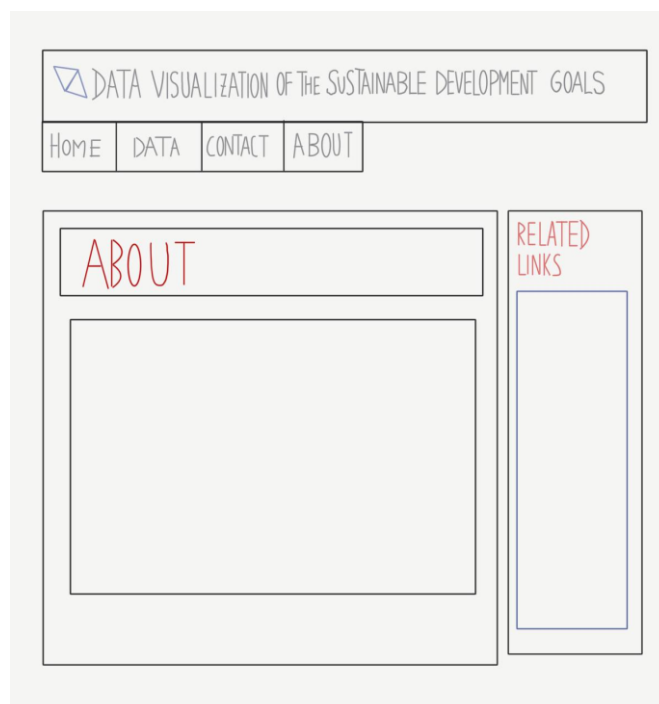
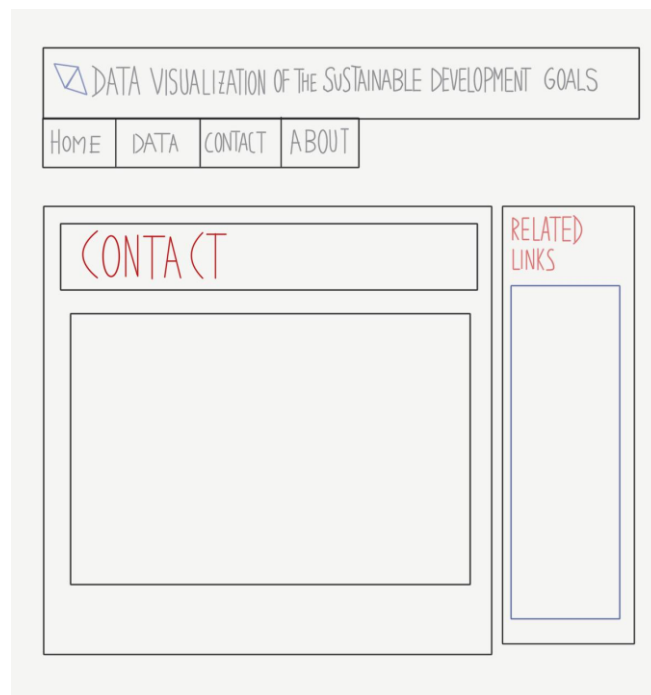
1. **Página principal.** Se accede pinchando en la pestaña “Home”, aunque es la “Landing Page” de la aplicación. Boceto:



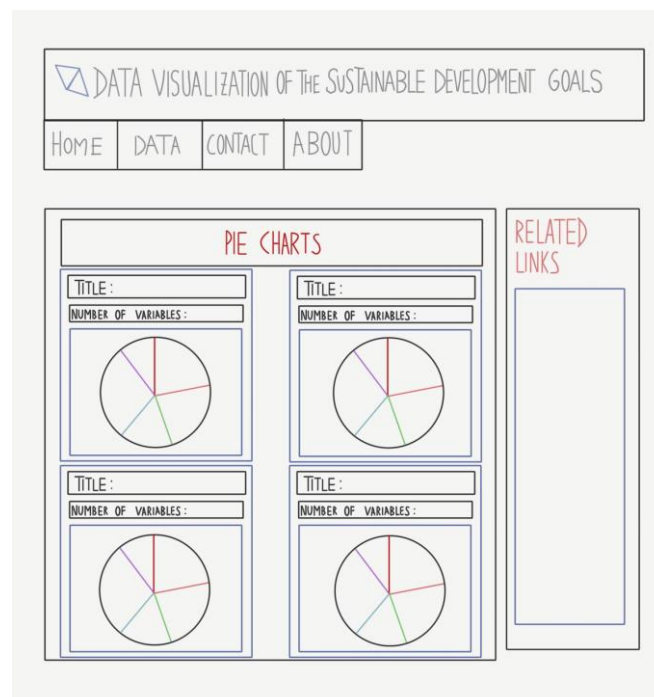
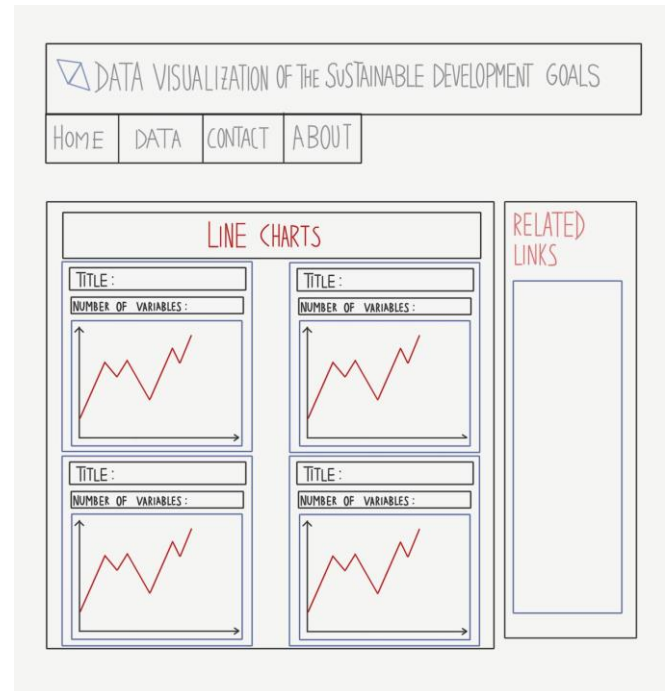
2. **Pestaña de selección de los datos.** Se accede pinchando en la pestaña “Data”. En ella se podrán seleccionar los datos que queremos ver representados en las gráficas. Se puedan seleccionar hasta cuatro archivos de datos. Boceto:



3. **Contacto e Información relacionada.** Se puede acceder pinchando en las pestañas “Contact” y “About”, respectivamente. Bocetos:



4. **Visualización de gráficas de barras, líneas y circulares.** Se puede acceder pinchando dentro de la pestaña “Home” en los apartados “Bar Chart”, “Line Chart” y “Pie Chart”, respectivamente. Bocetos:



4. Diseño

4.1. Diagrama de clases

El diagrama de clases es el siguiente:

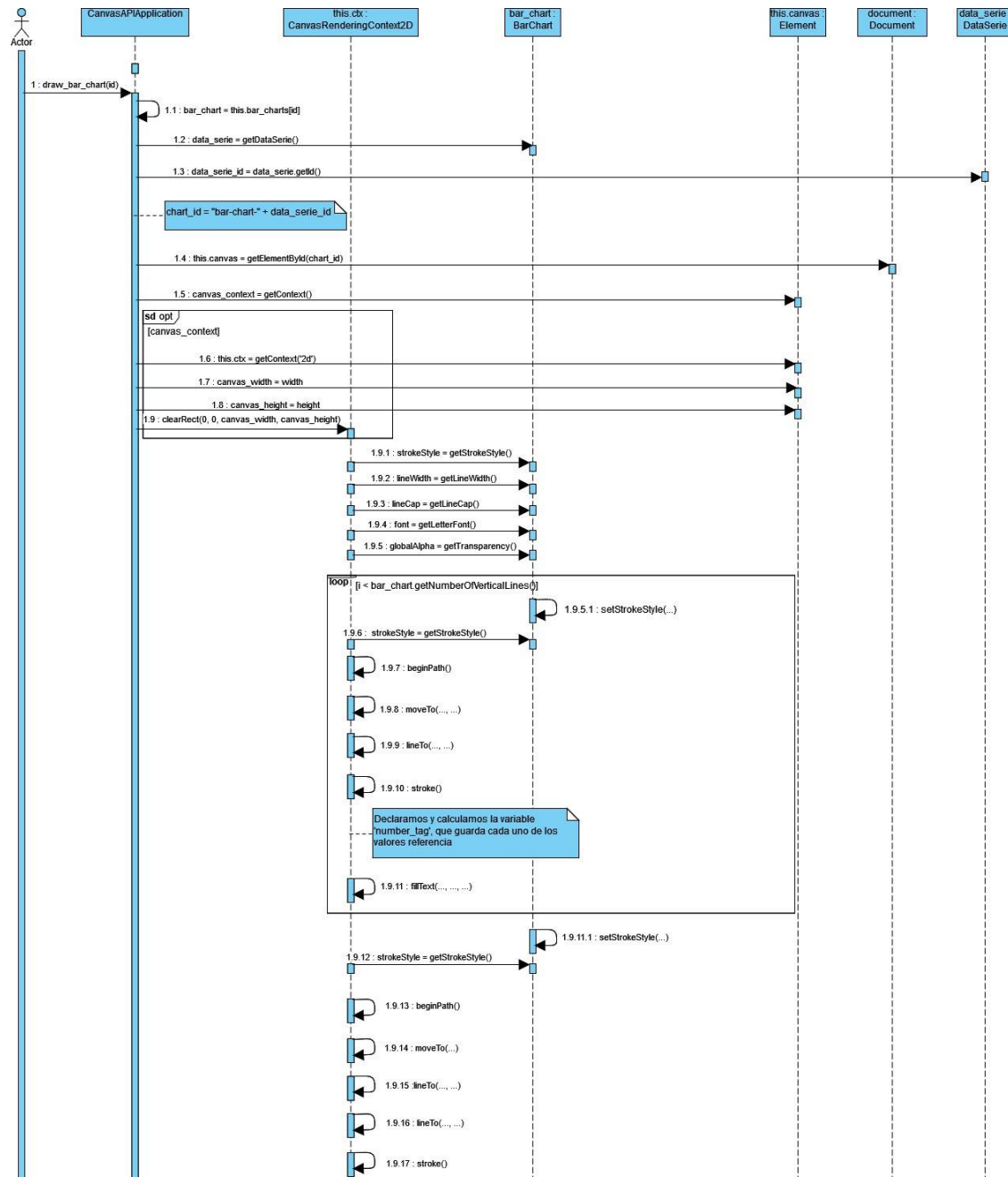
Las clases que aparecen en el diagrama de clases son las siguientes:

- *CanvasAPIApplication*. Clase encargada de gestionar y controlar la mayoría de las funcionalidades importantes de la aplicación. Posee un vector para almacenar las series de datos, un vector para las gráficas de barras, otro para las gráficas de líneas y otro para las gráficas circulares. También posee tres vectores para almacenar los identificadores de las gráficas que se van a dibujar en todo momento. Además, contiene tres atributos para establecer los siguientes identificadores de las series de datos y gráficas. Sus métodos se pueden dividir en cuatro tipos: *métodos de ocultación y visualizado de elementos*, *métodos de acceso a información*, *métodos de trazado de gráficas* y *métodos de configuración de atributos*.
- *DataSet*. Clase representativa de una serie de datos. Sus atributos están orientados a almacenar la información procesada o no procesada: mientras que la información no procesada se almacena toda en una cadena de caracteres, la información procesada se puede dividir en nombres de las variables, valores de la serie de datos, etiquetas de la serie de datos, etc.
- *Chart*. Clase representativa de una gráfica. Posee atributos de instancia que representan el estilo de la gráfica, así como los métodos pertinentes para modificar y obtener dichos atributos. También contiene métodos con el fin de insertar elementos gráficos en la aplicación. Contiene un método para calcular el color de cada una de sus secciones de manera aleatoria.
- *BarChart*. Clase representativa de una gráfica de barras. Esta clase hereda de la clase *Chart*. Contiene un conjunto extenso de atributos de clase con el fin de hacer más fácil el trazado de la gráfica. Redefine métodos para aprovechar el polimorfismo y ligadura dinámica.
- *LineChart*. Clase representativa de una gráfica de líneas. Contiene un conjunto extenso de atributos de clase con el fin de hacer más fácil el trazado de la gráfica. Redefine métodos para aprovechar el polimorfismo y ligadura dinámica.
- *PieChart*. Clase representativa de una gráfica circular. Contiene un conjunto extenso de atributos de clase con el fin de hacer más fácil el trazado de la gráfica. Redefine métodos para aprovechar el polimorfismo y ligadura dinámica.

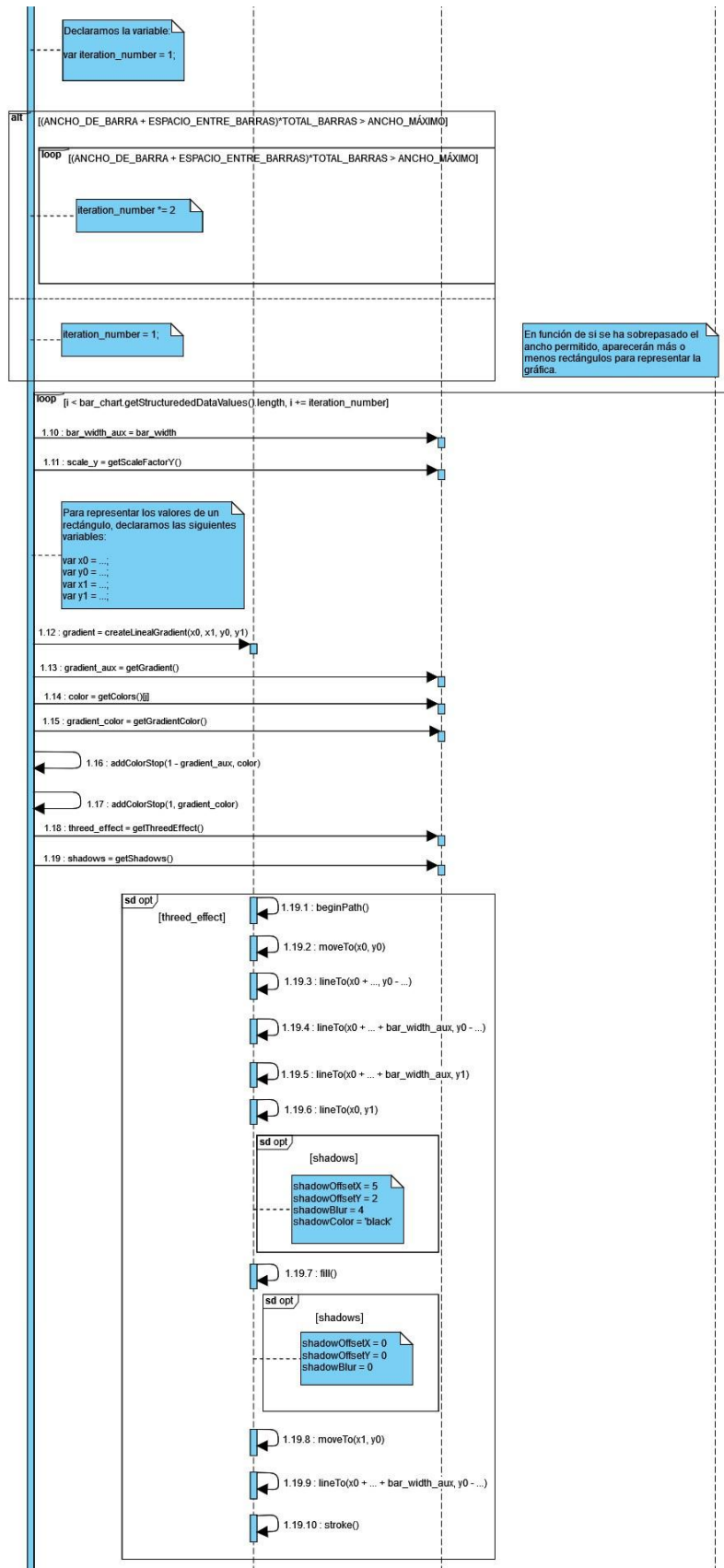
4.2. Diagramas de interacción

Trazado de gráficos de barras

El diagrama de interacción para el trazado de gráficas de barras es el siguiente:



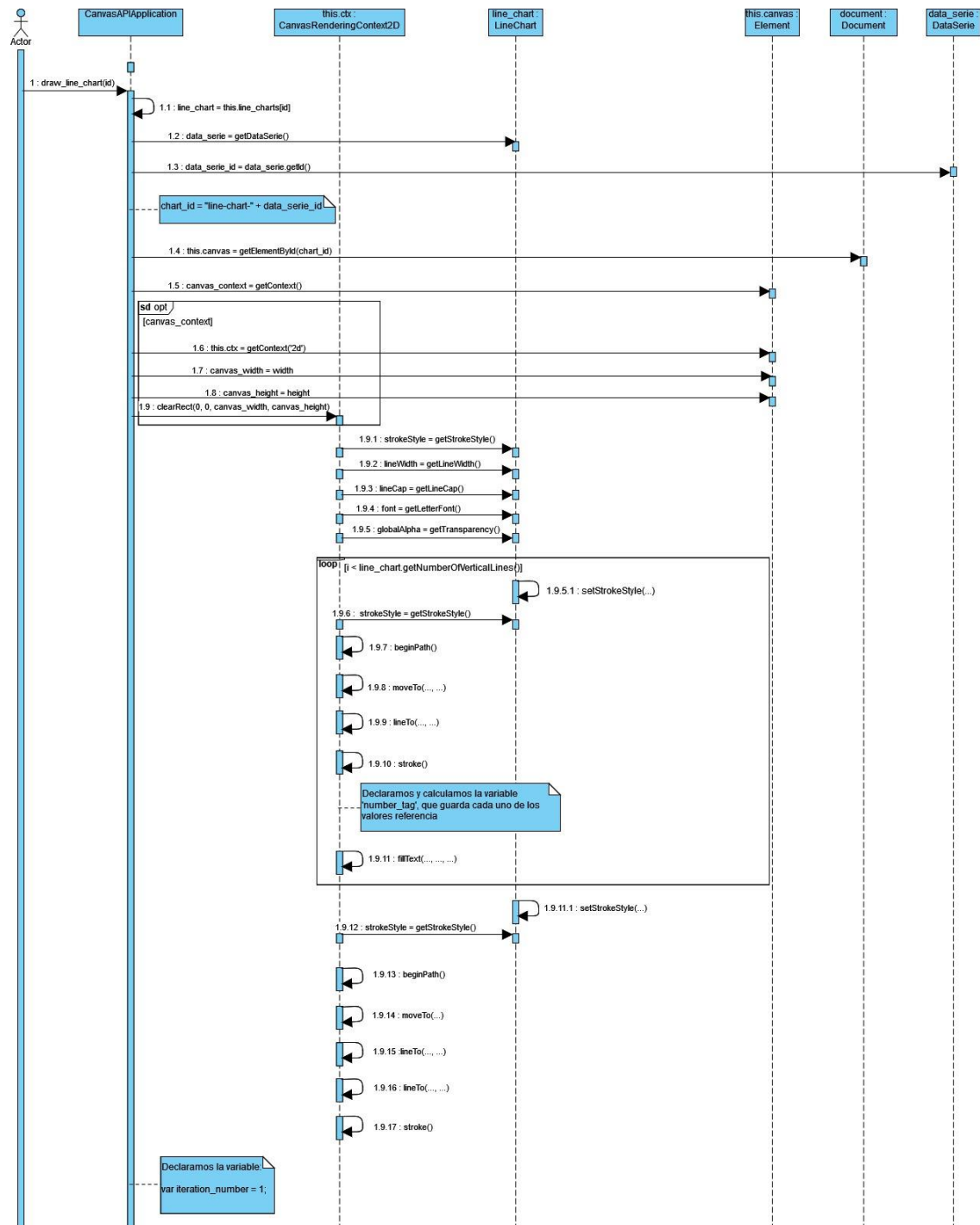
Continúa a continuación...



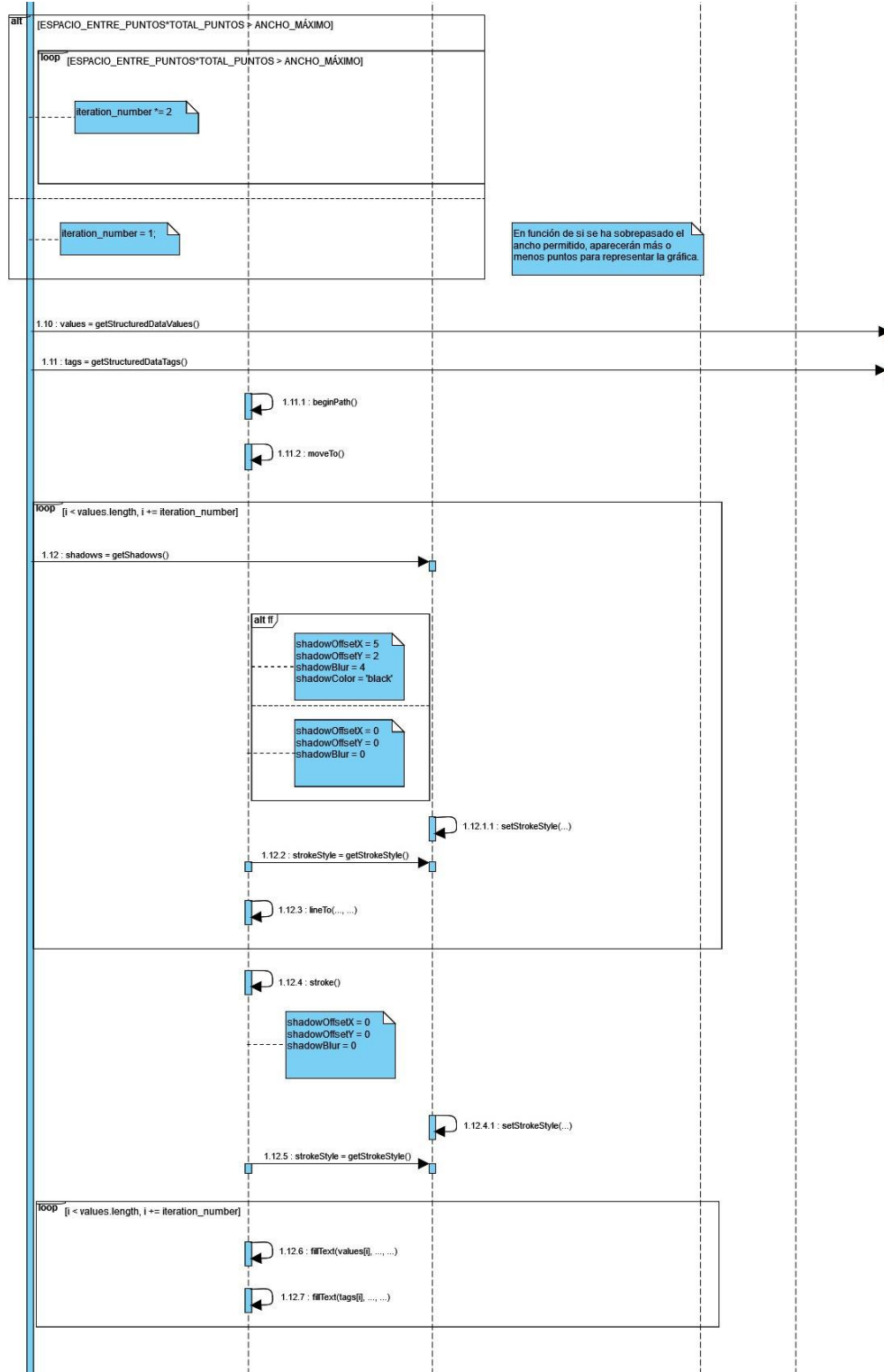
Continúa a continuación...

Trazado de gráficos de líneas

El diagrama de interacción para el trazado de gráficos de líneas es el siguiente:



Continúa a continuación...



Dicho proceso puede dividirse en distintas partes:

- *Trazado de los ejes de coordenadas.* Al igual que el gráfico de barras, es necesario trazar los ejes de coordenadas, con las mismas características y propiedad que el anterior gráfico.
- *Trazado de líneas.* Hacen alusión a los valores de la serie. También es necesario escalar el eje vertical utilizando el valor máximo de la serie y comprobar si se ha llegado al ancho máximo mediante el número total de valores.
- *Añadido de atributos de estilo.* Los atributos de estilo utilizados son grosor de línea, revestimiento de línea, sombras y opacidad. Posee menos atributos que el gráfico de barras debido al área reducida de la línea.

Trazado de gráficos circulares

El diagrama de interacción para el trazado de gráficas circulares es el siguiente:

4.3. Bocetos de las interfaces de usuario

Pestaña “Gráficas”

Pestaña “Datos”

Pestaña “Información”

Pestaña “Contacto”

4.4. Estructuras de datos fundamentales

Aquellas estructuras de datos necesarias para el desarrollo de la aplicación son:

- *Colección de gráficos de barras.* Está compuesto por todos los gráficos de barras preparados para su visualización. Cada uno tiene asociado una serie de datos.
- *Colección de gráficos de líneas.* Formado por todos aquellos gráficos de líneas listos para ser visualizados. Todos ellos tienen una serie de datos asociada.
- *Colección de gráficos circulares.* Constituido por todos los gráficos circulares habilitados para su visualización. Cada gráfico individual se encuentra asociado con una serie de datos.
- *Colección de identificadores de gráficos de barras a trazar.* Colección compuesta por los identificadores de aquellos gráficos de barras que están siendo visualizadas. No todos gráficos disponibles asociados a una serie de datos son visualizados a la vez, es el usuario el que elige cuáles quiere visualizar.
- *Colección de identificadores de gráficos de líneas a trazar.* Estructura formada por los identificadores de aquellos gráficos de líneas que pueden ser visualizados.
- *Colección de identificadores de gráficos circulares a trazar.* Se encuentra constituido por los identificadores de los gráficos circulares que son susceptibles de ser visualizados.
- *Colección de series de datos.* Constituido por aquellas series de datos cargadas por el usuario.

4.5. Desarrollo de algoritmos no triviales

Algoritmo de optimización de elementos a visualizar

El espacio utilizado para el mostrado de elementos gráficos es reducido y limitado, de forma que un algoritmo con el objetivo de perfeccionar esta funcionalidad era fundamental. El algoritmo tiene dos variantes en función de su uso:

- *Algoritmo para optimizar el espacio del gráfico de barras.* El algoritmo recoge aquellas modificaciones necesarias para optimizar el espacio de dicho gráfico. Consta de un proceso con tres partes diferenciadas:
 - **Primera etapa.** Se reducirá el ancho de cada barra y el espacio entre estas todo lo que sea necesario. En caso de que, al final de una iteración, se compruebe que es posible mostrar todos los elementos, el algoritmo salta a la siguiente etapa. De lo contrario, el algoritmo continúa.
 - **Segunda etapa.** Se reducirá el tamaño del texto utilizado para indicar los valores y las etiquetas. En cada iteración se comprueba si la optimización a finalizado o no.
 - **Tercera etapa.** Tras reducir el ancho de barra, el espacio entre estas y/o el tamaño del texto, es posible que los textos se solapen entre sí. Si este hecho ocurre, se muestran solo parte del texto, de forma que no se solape y pueda visualizarse correctamente.
- *Algoritmo para optimizar el espacio del gráfico de líneas.* Dicho algoritmo especifica aquellas modificaciones que tienen como objetivo optimizar el espacio en un gráfico de líneas. El proceso dispone de tres etapas:
 - **Primera etapa.** Progresivamente, se irá reduciendo el espacio entre cada punto todo lo que sea necesario. En caso de que no sea necesario continuar con la optimización, se concluye esta etapa y se procede a la siguiente.
 - **Segunda etapa.** Esta etapa disminuye el tamaño de los textos pertenecientes a los valores y las etiquetas todo lo que sea necesario. Mientras sea necesario, el algoritmo reduce el texto en cada iteración.
 - **Tercera etapa.** La última etapa comprueba si los textos de los valores o de las etiquetas se solapan. De ser así, se mostrarán solo algunos de ellos.

5. Implementación

5.1. Descripción de las clases

Clase ‘CanvasApiApplication’

Esta clase encapsula todas las funcionalidades principales de la aplicación, tales como la creación o borrado de series de datos y gráficas. En este caso, los **atributos de clase** son los siguientes:

- Atributo *HEIGHT_PIXELS*. Indica la altura, en píxeles, que tendrá el texto de cada una de las gráficas por defecto.
- Atributo *CORRECTOR_WIDTH*. Factor para corregir la anchura de cada una de las barras del gráfico de barras utilizando la anchura del texto.

Los **atributos de instancia** son los siguientes:

- Atributo *canvas*. Atributo utilizado para almacenar un elemento «canvas». Este es un elemento HTML utilizado principalmente para el trazado de gráficos. Almacena un objeto de la clase *Element*.
- Atributo *ctx*. Almacena el contexto de renderizado 2D para el trazado de elementos en «canvas». Es adquirido mediante el método *getContext('2d')* de este atributo. Almacena un objeto de la clase *CanvasRenderingContext2D*.
- Atributo *fileReader*. Atributo utilizado para almacenar un objeto de la clase *FileReader*. Este es utilizado para la lectura de ficheros.
- Atributo *data_series*. Vector utilizado para almacenar las series de datos creadas. Los elementos son objetos de la clase *DataSerie*.
- Atributo *data_serie_next_id*. Atributo utilizado para almacenar el identificador de la siguiente serie de datos una vez sea creada. El atributo es un entero.
- Atributos *bar_charts*, *line_charts* y *pie_charts*. Vectores utilizados para almacenar los gráficos de barras, de líneas y circulares, respectivamente, creados. Los elementos son objetos de las clases *BarChart*, *LineChart* y *PieChart*, respectivamente.
- Atributos *bar_charts_to_draw*, *line_charts_to_draw* y *pie_charts_to_draw*. Vector utilizado para almacenar los identificadores de los gráficos de barras que van a ser visualizados. Los elementos son números enteros.
- Atributos *bar_chart_next_id*, *line_chart_next_id* y *pie_chart_next_id*. Atributos utilizados para almacenar los identificadores de los siguientes gráficos de barras, líneas y circulares, respectivamente, una vez sean creados. Los atributos son enteros.

Los métodos de esta clase se pueden clasificar en los siguientes tipos:

- Métodos relacionados con la ocultación y la aparición de elementos. Estos se encargan de mostrar u ocultar elementos de la aplicación en función de las acciones del usuario. Ejemplos de estos métodos son *showHome()* o *showData()*.
- Métodos relacionados con la carga de ficheros. Estos tienen la función de la obtención de ficheros, ya sea seleccionándolos directamente o mediante un *drag and drop*. Ejemplos de estos métodos son *insertManager(file)* o *dropManager(event)*.
- Métodos relacionados con la creación o borrado de gráficos y series de datos. Estos desempeñan la función de insertar gráficos y series de datos, así como borrar las mismas, en función de las peticiones del usuario. Ejemplos de estos métodos son *submitChart()* o *removeDataSerie(id)*.
- Métodos relacionados con el trazado de gráficos. Estos desempeñan la función principal de la aplicación. Para cada gráfico se utiliza un «canvas» y un «ctx» (contexto) en particular.

Clase ‘DataSerie’

Clase ‘Chart’

Clase ‘BarChart’

Clase ‘LineChart’

Clase ‘PieChart’

5.2. Descripción de los métodos

6. Conclusiones y vías futuras

Bibliografía

- [1] The HTML5 Canvas Handbook.
URL: <https://bucephalus.org/text/CanvasHandbook/CanvasHandbook.html#introduction>
- [2] HTML Canvas Deep Dive: A Travelogue.
URL: <https://joshondesign.com/p/books/canvasdeepdive/title.html>
- [3] WebGL: Low-level 3D graphics API based on OpenGL ES
URL: <https://www.khronos.org/api/webgl>
- [4] Computer Graphics: Principles and Practice
- [5]
- [6] Craig Larman. «Agile and Iterative Development: A Manager's Guide». (2003)

https://developer.mozilla.org/es/docs/Web/API/Canvas_API/Tutorial
<https://developer.mozilla.org/en-US/docs/Web/API/CanvasRenderingContext2D>

Ken Schwaber. Agile Project Management with Scrum. Microsoft Professional. 2004