UNIVERSIDAD NACIONAL DE LA PLATA

FACULTAD DE INFORMÁTICA

TESINA DE LICENCIATURA

TÍTULO: Análisis y prototipado de un componente de visualización para usuarios finales en el marco de una herramienta de autor para recolección y análisis de datos. Un caso de estudio con perspectiva de género.

AUTORES: Rodriguez Pau Lola, Acha Altamiranda Pilar

DIRECTORAS: Patricia Bazan, Alejandra Lliteras

CARRERA: Licenciatura en Sistemas

Resumen

Este proyecto surge como extensión del proyecto "DEHIA: Una plataforma liviana para definir y ejecutar actividades con intervención humana basada en workflows", tesina de grado de Jose Arcidiacono [Arcidiacono, 2020], en el cual se presenta una aplicación web que permite a usuarios finales definir actividades que requieran de la intervención humana con el fin de realizar recolección y análisis de datos, así como una aplicación móvil para ejecutar dichas actividades.

Palabras Clave

Recolección de datos, Visualización, Interoperabilidad, Perspectiva de Género.

Conclusiones

En conclusión, se ha logrado el objetivo propuesto en la tesina, que consistió en diseñar e implementar una plataforma de visualización y análisis de datos para el Programa Institucional Contra la Violencia de Género en el ámbito de la Universidad Nacional de La Plata. La plataforma PIBAS propuesta permite visualizar y analizar los datos de manera interactiva y eficiente, y puede ser utilizada para tomar decisiones informadas en la lucha contra la violencia de género.

Trabajos Realizados

- Análisis de herramientas de visualización de datos.
- Análisis de la arquitectura existente en DEHIA.
- Se propuso un componente de visualización.
- Se diseñó la integración del componente propuesto con la arquitectura DEHIA.
- Implementación de un prototipo Web para visualización en base a la integración realizada.
- Se aplicó caso de estudio co-diseñado con la Dirección de Género y Diversidad de la Prosecretaría de DDHH-UNLP al prototipo Web.

Trabajos Futuros

- Recomendaciones de Gráficos Inteligentes
- Personalización Avanzada de Gráficos
- Colaboración en Tiempo Real
- Historial de Versiones de Gráficos
- Control de Acceso Basado en Roles

Fecha de la presentación: NOVIEMBRE 2023

UNIVERSIDAD
NACIONAL
DE LA PLATA

Facultad de Informática

Universidad Nacional de La Plata

Tesina de Grado

Licenciatura en Sistemas

PIBAS: Análisis y prototipado de un componente de visualización para usuarios finales en el marco de una herramienta de autor para recolección y análisis de datos. *Un caso de estudio con perspectiva de género.*

Autoras: Pilar Acha Altamiranda, Lola Rodriguez Pau

Directoras: Patricia Bazán, Alejandra B. Lliteras

ÍNDICE

Capítulo 1: Introducción **7**
1.1 Motivación 7
1.2 Objetivos 8
1.3 Sobre la organización de la tesina 9
Capítulo 2: Marco teórico **10**
2.1 Visualización de datos 10
2.2 Herramientas de visualización de datos 11
2.3 Ventajas de la visualización de datos 12
2.3.1 Mejora la comprensión de los datos 13
2.3.2 Facilita la toma de decisiones informadas 13
2.3.3 Permite la detección de patrones y tendencias 13
2.4 Recapitulación 14
Capítulo 3: Estado del arte y antecedentes **15**
3.1 Tableau 15
3.1.1 Autenticación 16
3.1.2 Carga de datos 16
3.1.3 Manipulación de datos 18
3.1.4 Visualizaciones 19
3.1.5 Exportar visualizaciones 20
3.2 CODAP (Common Online Data Analysis Platform) 21
3.2.1 Autenticación 21
3.2.2 Carga de datos 22
3.2.3 Manipulación de datos 23
3.2.4 Visualizaciones 23
3.2.5 Exportar visualizaciones 25
3.3 Google Data Studio 25
3.3.1 Autenticación 26
3.3.2 Carga de datos 26
3.3.3 Manipulación de datos 28
3.3.4 Visualizaciones 28
3.3.5 Exportar visualizaciones 30
3.4 Kibana 30
3.4.1 Autenticación 30
3.4.2 Carga de datos 31
3.4.3 Manipulación de datos 32
3.4.4 Visualizaciones 33
3.4.5 Exportar visualizaciones 34

3.5 Comparación de herramientas 34
3.6 Recapitulación 37
Capítulo 4: Plataforma PIBAS 38
4.1 Arquitectura Propuesta 38
4.1.1 DEHIA 39
4.1.2 Capa Cliente 39
4.1.3 Capa Servidor 41
4.1.4 Alternativas de la arquitectura 42
4.1.4.1 Enfoque distribuido 42
4.1.4.2 Enfoque integrado 43
4.1.4.3 Elección de la arquitectura 44
4.2 Implementación PIBAS 46
4.2.1 Integración de datos 47
4.2.1.1 Limpieza de datos 47
4.2.1.2 Integración de datos a DEHIA 48
4.2.2 Diseño de interfaz de usuario 49
4.2.3 Autenticación 53
4.2.3.1 Autenticación basada en sesiones 54
4.2.3.2 Autenticación basada en tokens 54
4.2.4 Tecnologías Utilizadas 55
4.2.4.1 Frontend: React.js 55
4.2.4.2 Backend: Node.JS con Express JS 56
4.3 Recapitulación 56
Capítulo 5: Caso de estudio 57
5.1 Problemática a resolver 57
5.2 Solución existente 59
5.3 Solución propuesta 60
5.3.1 DEHIA 60
5.3.2 PIBAS 63
5.4 Recapitulación 73
Capítulo 6: Conclusiones y trabajos futuros 74
6.1 Conclusiones 74
6.2 Contribuciones 75
6.3 Trabajos futuros 76
6.3.1 Recomendaciones de Gráficos Inteligentes 76
6.3.2 Personalización Avanzada de Gráficos 76
6.3.3 Colaboración en Tiempo Real 77
6.3.4 Historial de Versiones de Gráficos 77
6.3.5 Control de Acceso Basado en Roles 77
Referencias bibliográficas 78

ÍNDICE DE FIGURAS

Figura 3.1.1.1 - Hoja de trabajo vacía 15

Figura 3.1.2.1 - Carga de datos 16

Figura 3.1.3.1 - Creación de relaciones entre tablas 17

Figura 3.1.4.1 - Creación de una visualización 19

Figura 3.2.1.1 - Carga de hoja de trabajo 20

Figura 3.2.2.1 - Importar datos 21

Figura 3.2.4.1 - Documento con gráficos, tabla y ventana de texto 23

Figura 3.3.1.1 - Creación de informe 25

Figura 3.3.2.1 - Carga de fuente de datos 26

Figura 3.3.4.1 - Creación de visualizaciones 28

Figura 3.4.1.1 - Ingreso a la plataforma 30

Figura 3.4.2.1 - Cargar datos mediante un archivo 31

Figura 3.4.4.1 - Creación de una visualización 32

Figura 4.1 - Arquitectura implementada 37

Figura 4.1.2.1 - Administración de gráficos por actividad 39

Figura 4.2.1.2 - Administración de tableros 40

Figura 4.1.4.1.1 - Enfoque distribuido 42

Figura 4.1.4.2.1- Enfoque integrado en desarrollo 43

Figura 4.1.4.2.2 - Enfoque integrado en producción 43

Figura 4.2.1.1.1 - Subconjunto de resultados “UDA/Área interviniente” sin normalizar 47

Figura 4.2.1.1.2 - Subconjunto de resultados “Temática” sin normalizar 47

Figura 4.2.2.1 - Inicio 48

Figura 4.2.2.2 - Inicio de sesión 49

Figura 4.2.2.3 - Listado de actividades 50

Figura 4.2.2.4 - Creación de gráficos 50

Figura 4.2.2.5 - Creación de tablero 51

Figura 5.1.1 - Formulario de Google 56

Figura 5.2.1 - Hoja de cálculos de resultados 57

Figura 5.3.1.1 - Crear actividad en DEHIA 59

Figura 5.3.1.2 - Asociar tareas a una actividad 60

Figura 5.3.1.3 - Pregunta con respuesta cerrada o abierta 61

Figura 5.3.2.1 - Menú 62

Figura 5.3.2.2 - Selección de actividad en sección gráficos 63

Figura 5.3.2.3 - Valores disponibles para las medidas 64

Figura 5.3.2.4 - Indicadores con medida de desglose 65

Figura 5.3.2.5 - Tipos de gráficos disponibles 66

Figura 5.3.2.6 - Seleccionar el tipo de visualización para el eje x 66

Figura 5.3.2.7 - Seleccionar el tipo de visualización para el eje y 67

Figura 5.3.2.8 - Porcentaje por año de contacto inicial y vínculo institucional de la persona consultante 68

Figura 5.3.2.9 - Filtrar gráficos para la creación de un tablero 69

Figura 5.3.2.10 - Editar informe a descargar 70

Figura 5.3.2.11 - Informe 70

ÍNDICE DE TABLAS

Tabla 3.5.1 - Comparación de herramientas 35

Capítulo 1: Introducción

1.1 Motivación

Este proyecto surge como extensión del proyecto "DEHIA: Una plataforma liviana para definir y ejecutar actividades con intervención humana basada en workflows", tesina de grado de Jose Arcidiacono [Arcidiacono, 2020], en el cual se presenta una aplicación web que permite a usuariosfinales definir actividades que requieran de la intervención humana con elfin de realizar recolección y análisis de datos, así como una aplicación móvil para ejecutar dichas actividades.

Dado que esta plataforma pone el foco en el análisis de los datos, además de la recolección de los mismos, resulta de interés poder completar parte de la arquitectura de microservicios en la que se basa DEHIA con un visualizador que permita un análisis integral, eficaz yflexible, siguiendo la estructura yfilosofía de DEHIA, permitiendo de esta manera enriquecer la forma en la que la información es presentada.

La visualización de los datos recolectados y el conocimiento generado por las personas, permite traducir la información en gráficos y elementos visuales que facilitan a los usuariosfinales identificar patrones, extraer conocimiento significativo y distinguir las perspectivas que dicha información presenta. Así es como, a través de una buena visualización, el usuariofinal encuentra una manera accesible de interactuar con los datos y generar un posterior análisis, característica muy importante para construir una comunicación eficiente y concreta de los resultados obtenidos y el valor de los mismos. Lograr dichas visualizaciones puede resultar una tarea difícil para aquellas personas sin conocimientos específicos de este proceso y, particularmente, de las herramientas tecnológicas disponibles, es por eso que este componente de visualización se propone acortar la brecha entre la representación de los datos y sus usuariosfinales.

Paralelamente, se busca aplicar las visualizaciones en un proyecto institucional entre las UDAs-Áreas (Unidades de Atención y Áreas de Género, Diversidad y Derechos Humanos) y la Dirección de Género y Diversidad de la Prosecretaría de DDHH-UNLP.

Estas áreas apuntan al fortalecimiento de protocolos y estrategias de intervención, ante las situaciones que configuran violencia de género en el ámbito universitario. En este sentido, es de interés contar con herramientas que faciliten un posterior análisis que permita contribuir a mejorar el seguimiento y la evaluación de la política institucional de género.
El equipo interdisciplinario de las UDAs trabaja en la importancia de distinguir que "violencias por razones de género" no es sinónimo de "violencias contra las mujeres", siendo el primero un concepto más amplio, abarcativo e incluyente de las violencias por razones de género que se ejercen contra personas con identidades LGBTQI+ (lesbianas, gays, bisexuales, transexuales, queer, intersex y todos aquellos colectivos que no estén representados en dichas siglas). En ese sentido, el ejercicio de las violencias por razones de género implica una violación a los derechos humanos, y es por esto que creemos en la importancia de contribuir en el análisis y desarrollo de recursos institucionales que aporten a visibilizar la problematica y trabajar interdisciplinariamente en pos de sostener y construir una universidad libre de violencias.

Este proyecto institucional será tomado para validar la solución construida para esta tesina en torno a aportar valor a los datos recabados a través de las unidades, enfocándose en la visualización de datos dentro de un set de representaciones predefinidas.

1.2 Objetivos

La visualización de información es concebida como el conjunto de representaciones gráficas de datos que refuerza la interpretación de las personas, ayuda a trabajar con ellos de manera más eficiente y posibilitar el descubrimiento de patrones con ayuda de interacción y exploración visual. Las visualizaciones facilitan la comprensión y el análisis de grandes volúmenes de datos [Khalid & Zebaree, 2021].

Esta tesina de grado tiene por objetivo analizar, diseñar, desarrollar y poner en práctica una solución informática prototípica de código abierto, que permita a usuariosfinales

visualizar información generada en el marco de una herramienta de autor para la recolección de datos.

1.3 Sobre la organización de latesina

El resto del informe está estructurado de la siguiente forma:

- En el capítulo 2 se presenta el marco teórico que sustenta la tesina. Se refiere a temas relacionados con el enfoque, con la arquitectura y con el área de aplicación.
- En el capítulo 3 se discute el estado del arte en lo que respecta a plataformas para la visualización y el análisis de datos.
- En el capítulo 4 se desarrolla la plataforma PIBAS, propone una arquitectura y realiza la implementación actual de la arquitectura propuesta.
- En el capítulo 5 se describe el caso de uso que se eligió para instanciar el prototipo:*PROGRAMA INSTITUCIONAL CONTRA LA VIOLENCIA DE GÉNERO EN EL ÁMBITO DE LA UNIVERSIDAD NACIONAL DE LA PLATA*.
- En el capítulo 6 se detallan las conclusiones de la tesina donde además se presentan las contribuciones realizadas a partir de esta tesina y se presentan posibles trabajos futuros.

Capítulo 2: Marco teórico

La plataformización de la sociedad [Van Dijck et all., 2018] (Recreación, Comunicaciones, Relaciones interpersonales, Transporte, etc) hace que cada vez se recolecten más datos de las personas usuarias y del contenido que estas generan. El concepto de "Datificación" como el fenómeno por el cual se guardan datos de muchos más aspectos de los que el usuario cree [Mayer-Schönberger and Cukier, 2013]. Además, los formularios creados ad-hoc para analizar problemáticas sociales, como por ejemplo, la violencia de género, ponen de relieve la necesidad de dar soporte a la sociedad toda, para que pueda visualizar y analizar los datos recolectados, fundamentalmente para apoyar la toma de decisiones y la generación de nuevas políticas públicas (o la aplicación de las existentes) en los contextos más vulnerables.

Desde esta perspectiva, la visualización de datos mediante gráficos ayuda a la comunicación y permite descubrir información que los modelos y simulaciones pueden perder [Unwin, 2020], por ejemplo, valores atípicos y patrones [Unwin, 2020].

2.1 Visualización de datos

La visualización de datos es un área de estudio que se enfoca en la representación gráfica de datos y en la forma en que los usuarios pueden interpretar y comprender mejor la información presentada visualmente.

En la actualidad, los sistemas de visualización de datos son una parte integral de la mayoría de los sistemas de BI (*Business Intelligence*). La visualización de datos puede hacer visible lo invisible, ayuda a descubrir patrones y conexiones que de otra forma no serían visibles [Healy, 2018] . Además, los sistemas de visualización de datos también ayudan a los usuarios a identificar problemas potenciales y a descubrir oportunidades que de otra manera podrían pasar desapercibidas.

La visualización de datos ha demostrado ser una herramienta valiosa en el análisis y comunicación de información compleja. Las visualizaciones pueden incluir gráficos de barras, líneas, áreas, mapas de calor, diagramas de dispersión y muchos otros tipos de

gráficos y diagramas. Cada tipo de visualización es adecuado para diferentes tipos de datos y puede ser utilizado para resaltar diferentes patrones y tendencias.

Además, las herramientas de visualización de datos actuales están diseñadas para ser intuitivas y fáciles de usar, lo qué permite que usuarios sin conocimientos técnicos puedan crear visualizaciones y explorar los datos de manera efectiva. Algunas herramientas también ofrecen colaboración y compartición en línea para que los usuarios puedan trabajar en paralelo en la visualización y el análisis de datos.

En resumen, la visualización de datos es una técnica importante para el análisis y la comunicación de información compleja, permitiendo a los usuarios ver patrones y tendencias en grandes conjuntos de datos y tomar decisiones informadas. Las herramientas de visualización de datos modernas están diseñadas para ser intuitivas y fáciles de usar, lo que permite a los usuarios explorar y crear visualizaciones personalizadas sin la necesidad de conocimientos técnicos avanzados.

2.2 Herramientas de visualización de datos

Una buena visualización debe ser "clara, precisa, eficiente y estética" [Tufte E. R. , 2001]. Además, una buena visualización tiene que ser capaz de comunicar una historia con los datos. Para lograr estos objetivos, los diseñadores de visualización utilizan una amplia variedad de herramientas y técnicas, desde gráficos de barras y gráficos de líneas hasta mapas de calor, entre otros.

En el campo de la visualización de datos, existen diversas herramientas que ofrecen distintas funcionalidades para analizar y presentar información de manera gráfica e interactiva. También, esas herramientas tienen sus propias características y beneficios, lo que permite a los usuarios elegir la que mejor se ajuste a sus necesidades y objetivos. A continuación, nombraremos algunos ejemplos.

Tableau es una plataforma de visualización de datos líder en el mercado, que permite a los usuarios crear visualizaciones interactivas y tableros personalizados. Con Tableau, es posible conectar a diferentes fuentes de datos, lo qué incluye bases de datos, hojas

de cálculo y aplicaciones en la nube. La plataforma también cuenta con herramientas de análisis avanzadas, como análisis de series de tiempo y modelos predictivos.

CODAP es una herramienta gratuita y de código abierto diseñada específicamente para su uso en educación. Con CODAP, es posible explorar y visualizar datos de manera interactiva, lo que ayuda a los estudiantes a comprender mejor los conceptos estadísticos y matemáticos. La plataforma se integra con diferentes fuentes de datos, incluyendo hojas de cálculo y bases de datos, y ofrece herramientas de análisis y visualización de datos básicas.

Google Data Studio es otra herramienta de visualización de datos popular y gratuita, que permite a los usuarios generar informes personalizados y visualizaciones interactivas. La plataforma se integra con distintas fuentes de datos, incluyendo bases de datos SQL, Google Analytics y Google Ads. Además, Google Data Studio ofrece opciones de colaboración, lo que permite a los equipos trabajar juntos en la creación de informes y visualizaciones.

Por último, Kibana es una herramienta de visualización de datos de código abierto, desarrollada por Elastic, que se utiliza principalmente para la visualización de datos en tiempo real. Con Kibana, es posible conectar a diferentes fuentes de datos, incluyendo Elasticsearch, logstash y beats, lo que permite analizar y visualizar datos de logs, métricas y otros tipos de información en tiempo real. Kibana también cuenta con opciones de análisis avanzadas, como la detección de anomalías y la agregación de datos.

2.3 Ventajas de la visualización de datos

La visualización de datos es una herramienta poderosa para la exploración, el análisis y la comunicación de información compleja. Las principales ventajas de la visualización de datos y cómo puede mejorar la toma de decisiones informadas:

2.3.1 Mejora la comprensión de los datos

La visualización de datos es una forma efectiva de representar datos y hacer que los usuarios los comprendan mejor [Healy, 2018]. La visualización de datos puede ayudar a los usuarios a detectar patrones y tendencias en los datos, lo que puede ser difícil de ver en una tabla o un informe escrito.

La visualización de datos permite a los usuarios ver la información de una manera que es más natural y fácil de entender. Al mostrar los datos de una manera visual, la visualización puede hacer que la información sea más accesible y atractiva para los usuarios [Wilkinson, 2013].

2.3.2 Facilita la toma de decisiones informadas

Las visualizaciones de datos también pueden ayudar a los usuarios a tomar decisiones informadas. Los cuadros de mando (dashboards) pueden proporcionar una vista rápida y clara del rendimiento y los problemas del negocio. Los cuadros de mando pueden mostrar información clave de una manera visual y fácil de entender, lo que puede ser especialmente útil para los ejecutivos y gerentes que necesitan tomar decisiones rápidas basadas en datos [Few, 2013].

La visualización de datos puede ayudar a contar historias convincentes y persuasivas. Al presentar datos de una manera visualmente atractiva y fácil de entender, la visualización de datos puede ayudar a los usuarios a comunicar mejor sus ideas y persuadir a su audiencia [Knaflic, 2023].

2.3.3 Permite la detección de patrones y tendencias

La visualización de datos también puede ser útil para detectar patrones y tendencias en los datos. Puede ayudar a los usuarios a identificar patrones en los datos que de otra manera serían difíciles de ver. Al mostrar los datos de una manera visual, los usuarios pueden detectar patrones y tendencias en los datos más fácilmente, lo que puede ser útil para la toma de decisiones informadas [Tufte, 2021].

La visualización de datos puede ser especialmente útil para explorar grandes conjuntos

de datos. Al utilizar técnicas de visualización de datos, los usuarios pueden explorar grandes conjuntos de datos de manera más eficiente y detectar patrones y tendencias que de otra manera podrían pasar desapercibidos [Sosulski, 2018].

2.4 Recapitulación

En este capítulo hemos presentado una introducción al campo de la visualización de datos, incluyendo sus características, técnicas, herramientas y desafíos. Esta comprensión proporciona una base sólida para la creación de visualizaciones efectivas que pueden ser utilizadas para informar y guiar la toma de decisiones informadas.
En el siguiente capítulo se analizarán algunas de las plataformas existentes y luego se realizará una comparación de sus características y funcionalidades.

Capítulo 3: Estado del arte y antecedentes

Entre las alternativas de soluciones para la visualización y el análisis de datos mencionadas en el Capítulo 2, se destacan por suflexibilidad y potencial de construcción aquellas que permiten a usuariosfinales visualizar información sin necesidad de tener conocimientos específicos de este proceso. Algunas de ellas tienen como objetivo satisfacer las necesidades de distintos rangos de dominios, como área de negocios o educativos, coincidiendo sin embargo en proveer la interfaz necesaria para que diferentes conjuntos de datos generen visualizaciones de calidad. En este capítulo se analizarán cuatro ejemplos de este tipo de plataformas: 1-Tableau, 2-Google Data Studio, 3-Codapp, 4-Kibana y 5-Power Bi. A continuación, se describen las cuatro plataformas mencionadas, analizando el tipo de licencia, el tipo de versión y sus características, importación de datos, manipulación de los mismos y grados de visualización, para luego abordar la comparación entre ellas.

3.1 Tableau

Tableau[1] es una plataforma de creación de gráficos y tableros para el análisis y visualización de datos. Fue desarrollada en la Universidad de Stanford de California[2], a partir de un proyecto de ciencias de la computación (2003) que tenía como objetivo mejorar elflujo de análisis y la accesibilidad de los datos para los usuariosfinales a través de un visualizador.

Es de uso comercial, es decir sus productos son pagos, y ofrece una aplicación de escritorio (Tableau desktop) y una aplicación web (Tableau Public) a través de un servicio en la nube, con la posibilidad de solicitar, en el primer caso, una licencia gratuita de 1 año para estudiantes, y en el otro una prueba gratuita de días.

Tableau es una herramienta para la visualización de datos de propósito general, aunque también contiene características de visualizadores especializados en, por ejemplo, mapas temáticos, datos en función del tiempo, depuración de datos y estadísticas focalizadas.

[1] https://www.tableau.com/
[2] https://www.stanford.edu/

Para el relevamiento de los ejes presentados a continuación, se utilizó la versión de prueba gratuita de Tableau Public.

3.1.1 Autenticación

Los usuarios deben crear un perfil en la plataforma para poder utilizar la prueba gratuita que ofrece la versión web. Para el registro se pide ingresar un nombre, mail y contraseña, y luego validarlo a través de una url enviado al correo registrado.

Una vez autenticados, los usuarios interactúan con la plataforma creando un espacio/hoja de trabajo (Figura 3.1.1.1) donde tendrán acceso a todas las herramientas de análisis de datos y generación de gráficos que ésta ofrece.

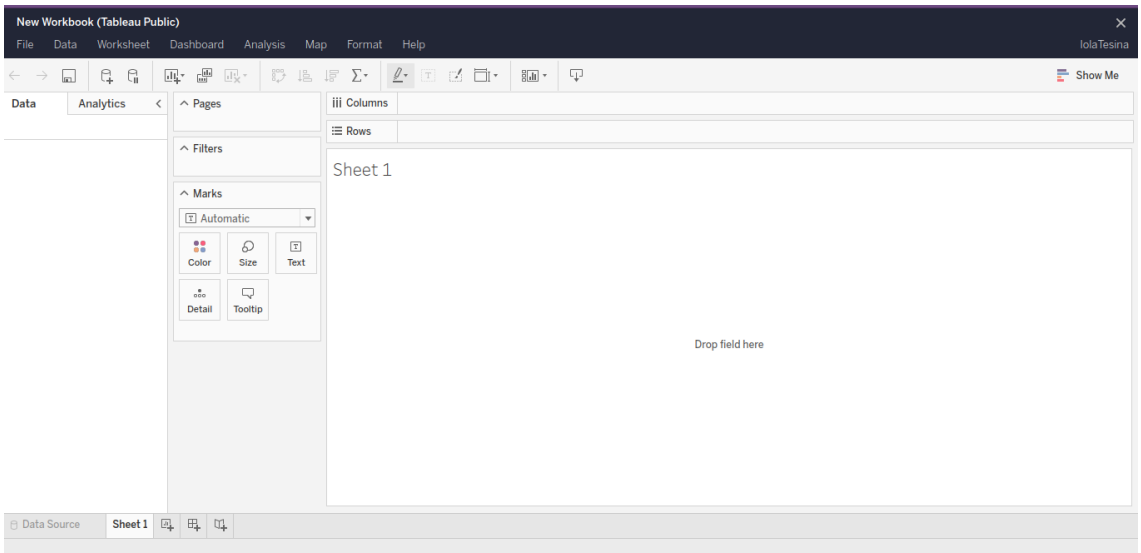


Figura 3.1.1.1 - Hoja de trabajo vacía

Como se puede ver en la Figura 3.1.1.1, la hoja de trabajo en Tableau contiene secciones para los datos y para las dimensiones de la visualización, sus configuraciones y la hoja donde se desplegará la construcción de la misma.

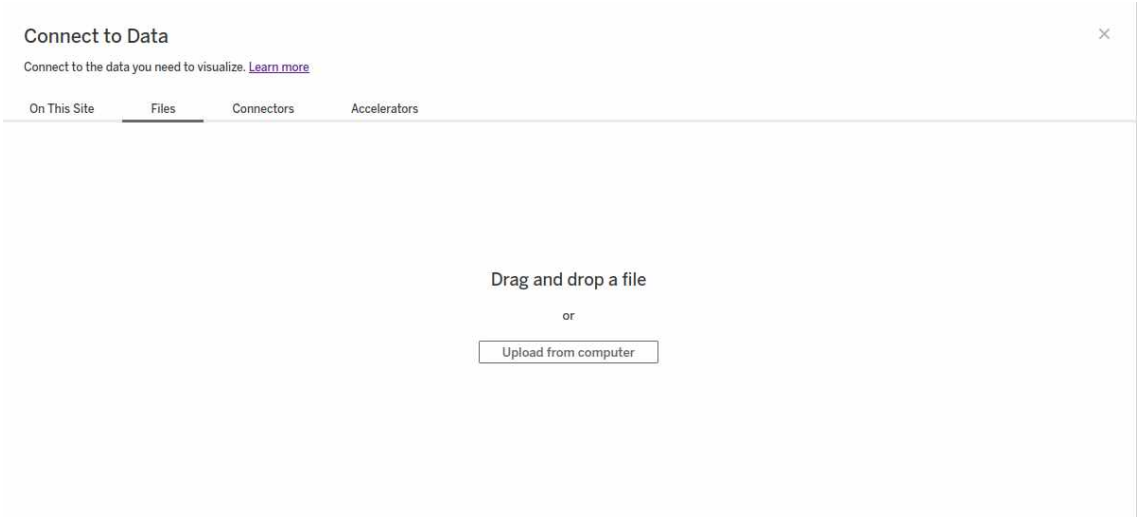
3.1.2 Carga de datos

Tableau ofrece 4 formas de cargar el conjunto de datos (Figura 3.1.2.1) con el que se desea trabajar las visualizaciones: archivos, aplicaciones externas, datos precargados en la sesión y datos de muestra. En el primer caso, la plataforma permite cargar datos a

través del sistema de archivos de la computadora soportando los formatos de libros de trabajo de excel (.xls, .xlsx, .xlsm), archivos de texto (.txt), archivos con delimitación de caracteres (.clsv), PDF (.pdf), archivos delimitados por tabulaciones (.tab, .tsv), archivos JSON (.json) y otros (.geojson, .topojson, .kml, .kmz, .zip).

La utilización de aplicaciones externas, llamadas "conectores", permite generar una conexión con datos alojados en una nube o servidores externos (Amazon Athena[3], Google Cloud SQL[4], Apache Drill[5], Azure SQL Database[6], IBM DB2[7], Microsoft Analysis Services[8], entre otros). Para cada una se debe cargar la información requerida en la conexión, por ejemplo nombre del servidor e información de registro. Los conectores soportados tienen la información necesaria para conectar a Tableau según el tipo de conector, de manera que se pueda configurar la fuente de datos en la plataforma. Una vez generada la conexión exitosamente, el usuario tiene la posibilidad de elegir, dentro de los datos alojados, el conjunto con el cual desea trabajar.

Tableau también nos ofrece utilizar conjuntos de datos cargados previamente en la sesión del usuario (en la hoja de trabajo actual o en otra, indistintamente), así como cargar tableros de datos pre-elaborados con datos de prueba o datos en la nube de una cuenta Salesforce[9], en caso de tenerla.



[3] https://aws.amazon.com/athena
[4] https://cloud.google.com/sql
[5] https://drill.apache.org/
[6] https://azure.microsoft.com/es-es/products/azure-sql/database/
[7] https://www.ibm.com/ar-es/products/db2-database
[8] https://docs.microsoft.com/en-us/analysis-services
[9] https://www.salesforce.com

Figura 3.1.2.1 - Carga de datos

Tableau tiene como límite para el conjunto de datos importado en la hoja de trabajo un tamaño máximo de 15 GB.

3.1.3 Manipulación de datos

Tableau ofrece una sección para el tratamiento de el/los conjunto/s de datos importados a la hoja de trabajo, siguiendo la estructura de una base de datos relacional.

Las columnas que representan cada dato son presentadas dentro de una tabla que modela el tipo de dato, nombre del campo que representa, tabla física donde se encuentra referida y nombre remoto del campo. Además, de cada columna/tupla, se permite modificar el tipo de dato a número decimal, número entero, fecha y hora, fecha, string, boolean, default o rol geográfico, y realizar acciones sobre el nombre del campo (duplicar, renombrar, copiar valores o esconder).

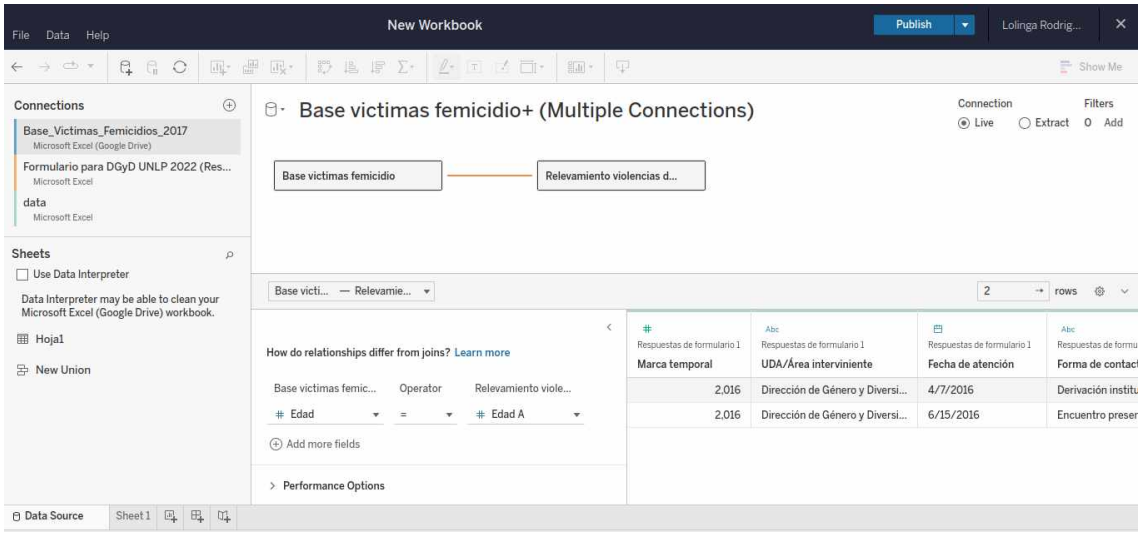


Figura 3.1.3.1 - Creación de relaciones entre tablas

Por otro lado, a través del recurso de drag-and-drop (arrastrar y soltar), se permite relacionar distintos conjuntos de datos importados indicando a través de qué campos se van a combinar y con qué operador se requiere construir la relación (Figura 3.1.3.1). La plataforma valida que los campos sean del mismo tipo y utiliza el concepto de relaciones para combinar las tablas, aunque también da la posibilidad de construirlas manualmente mediante joins (uniones).

3.1.4 Visualizaciones

Tableau ofrece la construcción de visualizaciones a través de los siguientes tipos de gráficos: gráfico de barras, gráfico de líneas, mapas, gráficos de densidad, gráficos de dispersión, gráfico de burbujas, gráfico de histograma, gráfico de balas (viñetas), tabla resaltada, mapa de árbol, diagrama de caja y gráfico de velas japonesas (de uso financiero). En cada uno de ellos, define una cantidad de dimensiones (valores cualitativos) y una cantidad de medidas (valores cuantitativos) mínimas necesarias que el subconjunto de datos debe cumplir para utilizar el tipo de gráfico.

La plataforma procesa los datos importados tomando las columnas como campos y asignando roles y tipos a cada uno de ellos. En particular, los tipos que cada campo puede tomar son número entero, cadena de texto, fecha, entre otros (mencionados en la Sección 3.1.3), y los roles se determinan en base a dos grandes categorías: dimensiones (valores cualitativos, pueden ser utilizados para categorizar, segmentar y profundizar el nivel de la información) y medidas (valores cuantitativos, son numéricos y permiten medir información). A su vez, ambas medidas se dividen en dos categorías: continuas (rango infinito) y discretas (rangofinito). Las más usadas comúnmente son las dimensiones discretas y las medidas continuas.

Para la administración de las visualizaciones, la hoja de trabajo contiene el listado de campos, las columnas yfilas que configurarán la visualización junto con características para personalizarla (color, etiquetas, tamaño, entre otras) y el listado de gráficos mencionados previamente, habilitados o deshabilitados según corresponda. Además, agrega información que puede ser útil para el usuario, como el total de valores para un campo.

Tableau permite seleccionar fácilmente los campos con los que se desea trabajar y asignarlos al listado de columnas o defilas, a través del recurso de drag-and-drop. En el mismo momento en el que se van configurando las columnas y lasfilas, Tableau construye la visualización más cercana posible al subconjunto de datos seleccionados (Figura 3.1.4.1) y habilita los gráficos que se permiten construir.

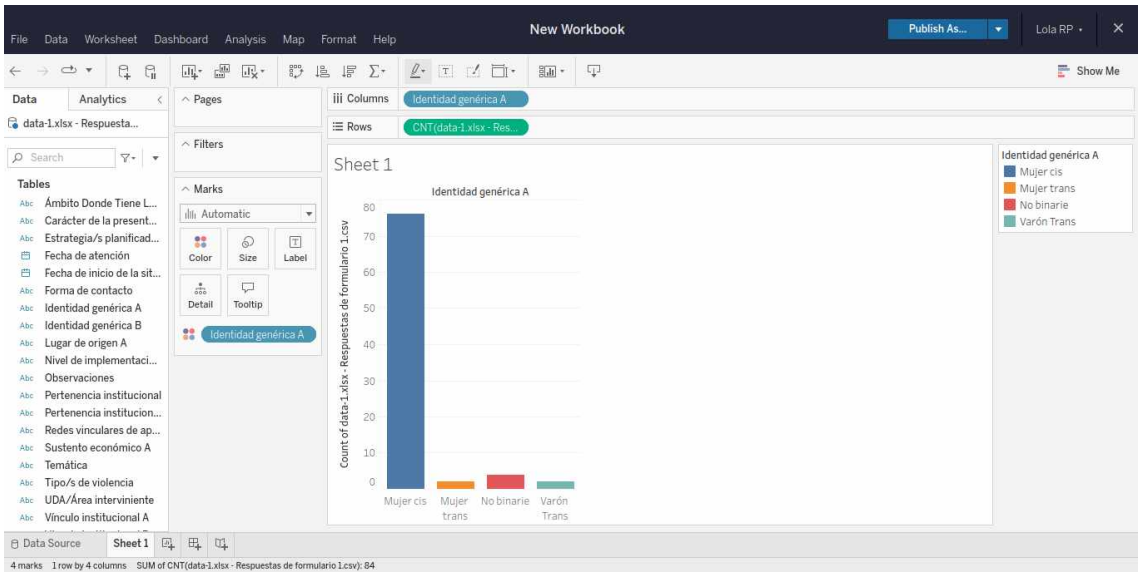


Figura 3.1.4.1 - Creación de una visualización

Luego del proceso de construcción previamente mencionado, la hoja de trabajo queda poblada por la visualización y sus configuraciones asociadas, como se puede ver en la Figura 3.1.4.1.

Además, Tableau permite la creación de tableros donde se pueden combinar visualizaciones, agregar etiquetas, mostrar distintos niveles de información.

3.1.5 Exportar visualizaciones

El usuario tiene la posibilidad de exportar las visualizaciones a varios formatos: imagen (.png), archivo separado por comas (.csv), que contiene el subconjunto de datos que comprenden la visualización, tabla cruzada en formato excel (.xlsx) o de archivo .csv, PDF (.pdf), presentación de powerpoint, donde cada hoja de trabajo se exporta como imagen a una diapositiva y hoja de trabajo de Tableau, con la condición de que la misma esté guardada (publicada en la plataforma).

3.2 CODAP (Common Online Data Analysis Platform)

CODAP[10] es un software gratuito de código abierto (licencia MIT[11]) para el análisis de datos. Fue construido bajo elfinanciamiento de la National Science Foundation [12], agencia gubernamental de los Estados Unidos, y diseñado especialmente para uso educativo. Tiene como objetivo brindar una plataforma web gratuita basada en exploración de datos y herramientas de visualización, y se encuentra disponible en inglés, chino, alemán, hebreo, español y turco.

CODAP se basa en documentos, lo cual facilita la importación de datos, y posee una interfaz de usuario basada en el recurso arrastrar y soltar, donde además se permite (con algunas restricciones) el*web scraping* [13].

3.2.1 Autenticación

La aplicación web no requiere la generación de un perfil de usuario. Al ingresar se presentan opciones para el documento sobre el cual se quiere trabajar: crear uno nuevo, abrir uno existente (desde Drive o el sistema de archivos local) o documentos de ejemplo cargados en la plataforma (Figura 3.2.1.1).

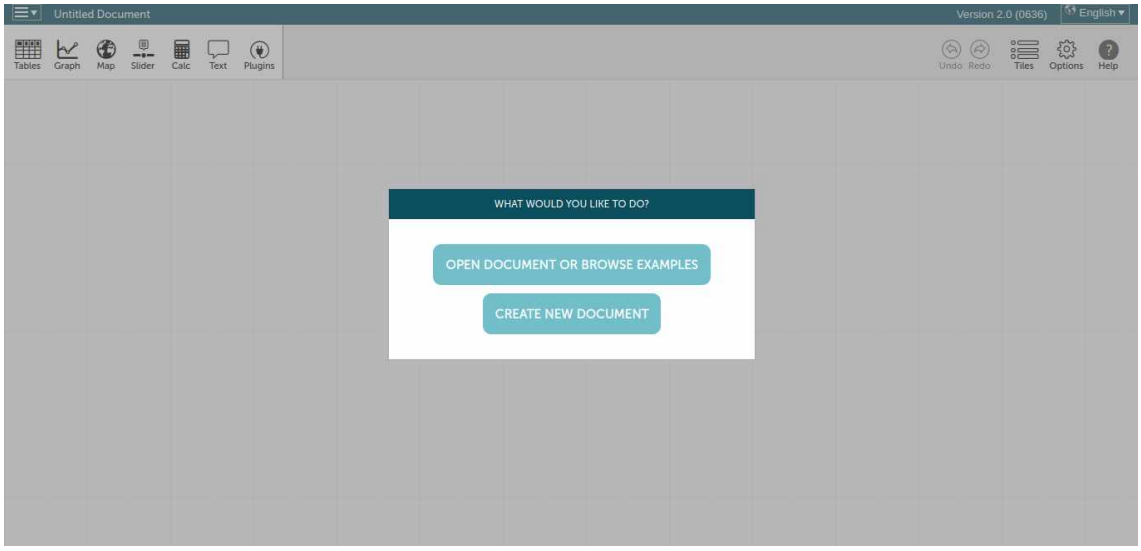


Figura 3.2.1.1 - Carga de hoja de trabajo

[10] https://codap.concord.org/
[11] https://opensource.org/licenses/MIT
[12] https://www.nsf.gov/
[13] *Web scraping*es una técnica utilizada mediante programas de software para extraer información de sitios web. https://es.wikipedia.org/wiki/Web_scraping

Dicho documento, es análogo a la hoja de trabajo de Tableau pero también a los tableros, ya que se pueden asignar varias visualizaciones al mismo.

3.2.2 Carga de datos

CODAP permite cargar datos manualmente, es decir definiendo las columnas e ingresando lasfilas una a una desde teclado, e importar datos desde un archivo local aceptando los formatos .csv, .json, .txt y .codap (Figura 3.2.2.1). También ofrece la posibilidad de importar datos desde una url, limitado a referencias cuyo formato sea procesable por la plataforma.

Las opciones para la carga de datos en CODAP resultan acotadas en relación a Tableau, por ejemplo, y otras herramientas de visualización más robustas. La estructura que deben llevar los datos para poder ser procesados por la plataforma es sensible, y se recomienda hacer un proceso previo de ajuste para evitar errores en la importación.

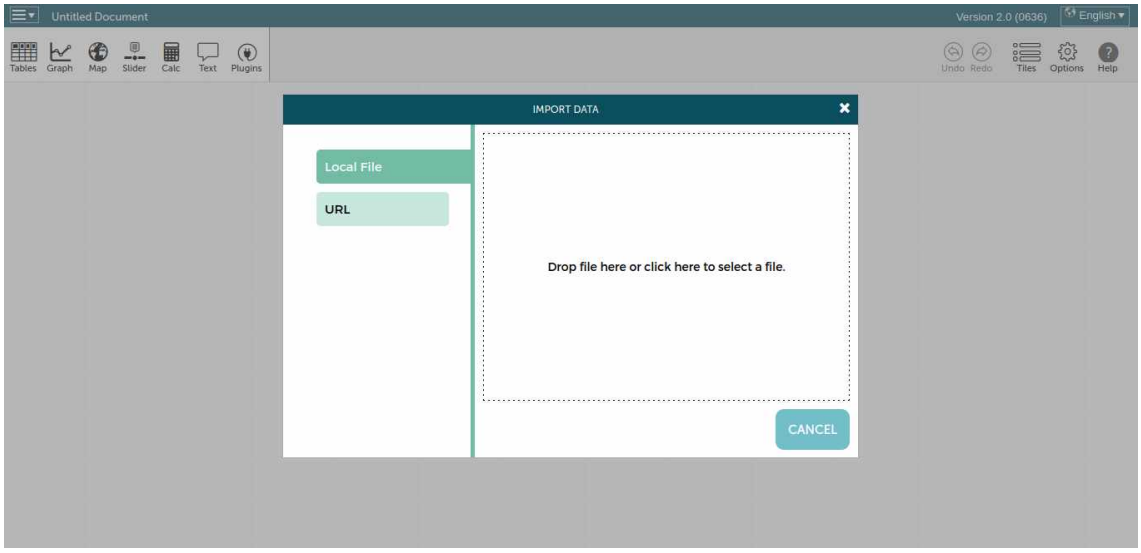


Figura 3.2.2.1 - Importar datos

CODAP tiene como límite para los conjuntos de datos a manejar en la plataforma una cantidad de 5000 casos, es decir el tamaño máximo de archivo que acepta CODAP está dado por la cantidad de casos (filas) en los que se traduzca.

3.2.3 Manipulación de datos

CODAP no contiene herramientas que permitan la manipulación de datos. Una vez realizada correctamente la importación, el conjunto de datos se carga en una tabla y, en cada columna, se permite modificar las propiedades del atributo (nombre, descripción, tipo, unidad, precisión y si es editable). Los tipos de datos que acepta CODAP son: categórico (que contiene propiedades cualitativas), numérico, fecha y hora, cualitativo (atributo numérico visualizado en medidas absolutas), límite (rangos, utilizado para visualizar mapas).

3.2.4 Visualizaciones

Las características que ofrece el visualizador CODAP son: tablas, gráficos, mapas, sliders, calculadora, texto y plugins. Esta última es parte de un enfoque de desarrollo que tiene CODAP, en el cual expone una API que permite desarrollar plugins para manipular datos y construir visualizaciones.

La creación de un gráfico inicia por defecto en un gráfico de dispersión, y se configura arrastrando variables a los ejes x e y. Además, soltando dentro del espacio del gráfico el encabezado de alguno de los campos que funcionan como eje, se le asigna un color a cada valor del dato en la visualización, mostrando el listado de referencias por debajo.

Los tipos de gráficos que ofrece CODAP a priori son acotados, aunque pueden extenderse mediante la utilización y/o desarrollo de plugins, y se basan en un gráfico de dispersión que visualiza las variables en puntos (Figura 3.2.4.1). En caso de tener una variable numérica en alguno de los dos ejes, se agrega la opción de agruparlos en contenedores o visualizarlos en barras. Dicha característica extiende el gráfico inicial de dispersión, aunque resulta poco intuitiva para el usuario a la hora de entender las opciones que presta el visualizador.

En caso de tener subconjuntos de datos con valores de latitud y longitud, CODAP permite representarlos dentro de mapas. También ofrece la posibilidad de visualizar la información según su frecuencia y configurar el grado de granularidad de la misma, así como categorizar con colores y referencias los valores para un atributo, al igual que en

los gráficos. Además, CODAP ofrece la generación de “sliders” que representan parámetros a través de números etiquetados. La potencialidad de esta herramienta recae en poder referenciar el valor en fórmulas y visualizar su comportamiento en los distintos rangos de valor que puede tomar.

CODAP también permite utilizar una calculadora, agregar ventanas de texto y emplear plugins desarrollados por fuera de la plataforma. Las ventanas de texto son útiles para asociar información a las visualizaciones que existan en el documento de trabajo (Figura 3.2.4.1). Además, dado que el desarrollo de CODAP pone el foco en su uso educativo, resulta de un recurso valioso para asociar herramientas pedagógicas que construyen conocimiento en la plataforma, como tutoriales o actividades de análisis de los datos. Por otro lado, la posibilidad de implementar plugins permite extender las herramientas de la plataforma.

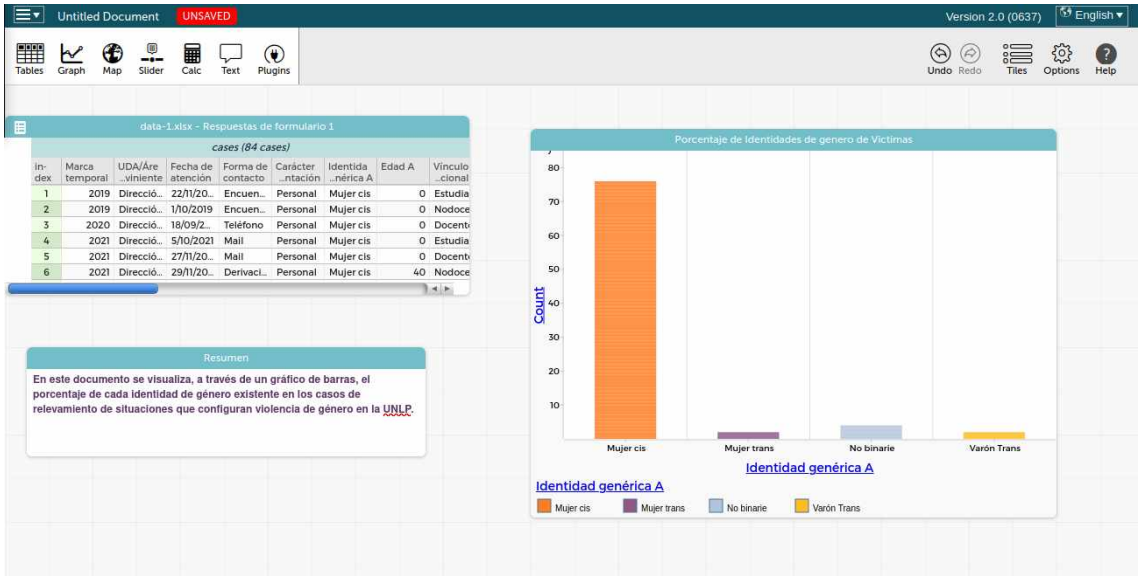


Figura 3.2.4.1 - Documento con gráficos, tabla y ventana de texto

Finalmente, el usuario puede construir una hoja de trabajo con visualizaciones, su conjunto de datos e información asociada.

3.2.5 Exportar visualizaciones

CODAP permite exportar las visualizaciones individualmente como una imagen en formato .png. En cuanto al documento, la plataforma ofrece el formato .codap para descargarlo localmente o guardarlo en una cuenta de Google drive y también genera la URL necesaria para embeber el mismo en una página web. Por último, permite exportar las tablas de datos en un archivo delimitado por comas (.csv).

3.3 Google Data Studio

Data Studio[14] es una herramienta web basada en la nube para la generación de informes de datos, con un diseño intuitivo que facilita la creación de tableros e informes personalizados. Es parte del conjunto de herramientas de Google Marketing Platform[15], que contiene varios productos con el objetivo de mejorar el marketing de pequeñas y grandes empresas, y hasta el momento se ofrece de forma completamente gratuita. Data Studio está disponible en 37 idiomas, incluidos el español, inglés y francés, entre otros.

La plataforma permite recopilar información de distintas fuentes de datos para la creación de gráficos, tablas y otras representaciones visuales. Además, a partir de las visualizaciones construidas, el usuario puede crear paneles de control en función de sus objetivos e informes a través de plantillas extensibles y personalizables. Por último, es posible dar u obtener acceso a trabajar colaborativamente con otros usuarios.
Google Data Studio está organizado en cuatro grandes bloques de funciones: integración (importar datos y mantener la estructura de los mismos, encargándose además de su almacenamiento), transformación (configurar las propiedades del conjunto de datos y procesarlas), visualización (instanciar datos en un recurso visual asociándole características) y uso compartido de datos (permitir el trabajo colaborativo con otros usuarios).

[14] https://datastudio.google.com
[15] https://marketingplatform.google.com/about/

3.3.1 Autenticación

Data Studio autentica y autoriza al usuario a través de su cuenta de Google. Al ingresar, se accede a la pantalla principal donde están todas las funcionalidades que ofrece la plataforma, incluida la generación de un espacio de trabajo, llamado informe (Figura 3.3.1.1).

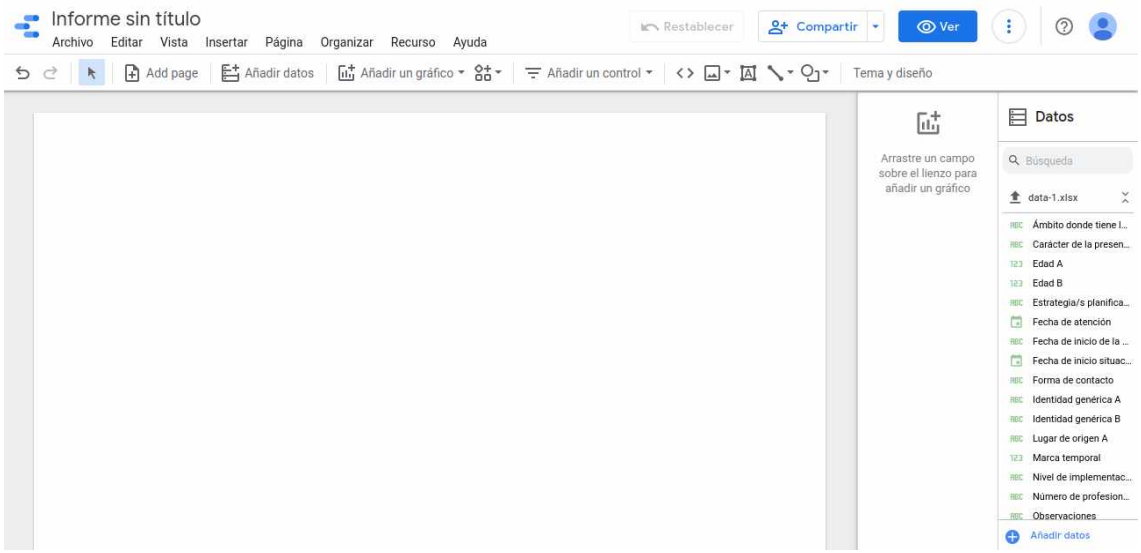


Figura 3.3.1.1 - Creación de informe

Mediante la creación de un informe, el usuario accede a una hoja de trabajo en la cual accede a sus datos y construye las visualizaciones.

3.3.2 Carga de datos

Para poder crear un informe, primero se deben definir las fuentes de datos con las que se desea trabajar (Figura 3.3.2.1).

Google Data Studio ofrece un amplio listado de herramientas de la familia de Google con las que crear conexiones y obtener información en tiempo real (como BigQuery[16], Cloud SQL[17], Google Analytics[18], Google Ads[19] o Google Cloud Storage[20] entre otras), y

[16] https://cloud.google.com/bigquery?hl=es
[17] https://cloud.google.com/sql?hl=es
[18] https://analytics.google.com
[19] https://ads.google.com
[20] https://cloud.google.com/storage?hl=es

también permite importar datos de numerosas fuentes externas como Mailchimp[21], Microsoft[22], Shopify[23], MySQL[24], PostgreSQL[25], redes sociales (LinkedIn[26], Facebook[27], etc.) y muchas otras fuentes. Asimismo, el usuario puede importar fuentes de datos propias cuyo formato no está restringido a priori, sin embargo debe respetar una estructura para que sea procesado por la plataforma correctamente. Data Studio describe con detalle en su documentación[28] cada punto de la estructura, donde los archivos deben contener datos tabulados, estar codificados en UTF-8, separados por coma y cuyasfilas contienen en la primera los encabezados yfinalicen cada una con un salto de línea.

Además, Data Studio trabaja con el concepto de*data set*(conjunto de datos) mediante el cual permite asociar diferentes fuentes de datos y crear una nueva fuente que alimente la generación de informes.

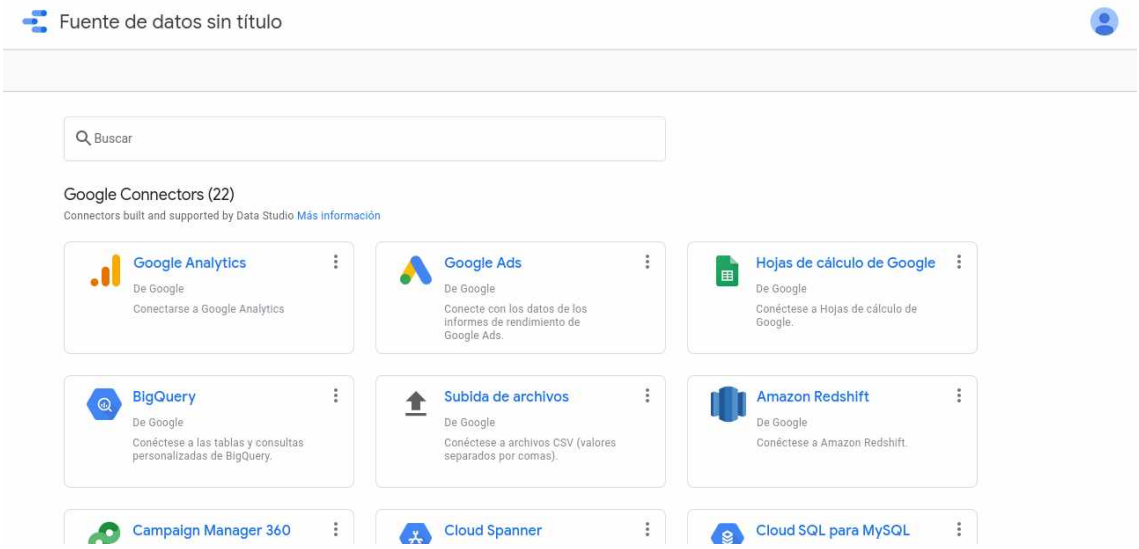


Figura 3.3.2.1 - Carga de fuente de datos

[21] https://mailchimp.com/es/
[22] https://www.microsoft.com/es-ar
[23] https://es.shopify.com/
[24] https://www.mysql.com/
[25] https://www.postgresql.org/
[26] https://www.linkedin.com/
[27] https://www.facebook.com
[28] https://support.google.com/datastudio/answer/7333350?hl=en&ref_topic=7333349#zippy=%2Cin-this-article

Data Studio define en la importación de conjuntos de datos un tamaño máximo de archivo de 100 MB para cada uno.

3.3.3 Manipulación de datos

Data Studio permite manipular los datos importados a la plataforma como una base de datos relacional, particularmente en SQL. Para ello, ofrece la posibilidad de configurar los campos modificando su tipo y/o su nombre, asociando agregaciones y agregando una descripción a cada uno, y a su vez construir, a partir de los existentes, nuevos campos y parámetros (campos que aceptan valores de entrada y los procesan con un tipo y un rango de valores).

El manejo de tipos se encuentra dentro de: numéricos (número, porcentaje o duración), texto, fecha y hora (con sus distintas variantes de formato), booleano, información geográfica (con sus distintos valores de representación), moneda (con todos los tipos existentes) y URL (url o imagen). Para las agregaciones se ofrece: total, media, recuento, recuento diferenciado, mínimo, máximo, mediana, desviación estándar y varianza.

3.3.4 Visualizaciones

Data Studio brinda al usuario un espacio de trabajo en el cual primero deberá cargar datos (Sección 3.3.2) para luego comenzar con la construcción de visualizaciones.

Data Studio posee un espacio de trabajo en el que se puede crear más de una visualización (Figura 3.3.4.1) y permite la creación de gráficos de tipo tabla y tabla dinámica (común, con barras o con mapa de calor para cada tipo), tarjeta de resultados (común o con números abreviados), serie temporal (común, minigráfico o suavizado), barras de tipo vertical u horizontal (común, apilado o apilado al 100% para cada tipo), circular (común o anillo), google maps (de burbujas, coroplético o de calor), mapa geográfico común, líneas (común, suavizadas o combinado común y combinado apilado), áreas (común, apiladas o apiladas al 100%), dispersión (común o en burbujas), bala, gráfico de rectángulos y, por último, de indicador (común o con intervalos).

También agrega la posibilidad de utilizar “visualizaciones comunitarias” que son complementos desarrollados externamente a Data Studio, es decir por otras y otros desarrolladores, análogo a los “plugins” mencionados previamente en CODAP (Sección 3.2).

Data Studio posee un menú lateralfijo en su hoja de trabajo donde se encuentran todos los campos disponibles de la fuente seleccionada y que permite, a través del recurso de drag-and-drop, configurar aquellos datos que van a servir al gráfico elegido. Para ello, la plataforma permite darle forma a través de dimensiones (valores categóricos que agrupan la información) y métricas (valores cuantitativos que brindan una escala numérica de los datos), así como dimensión por periodo, ordenación ascendente o descendente de los datos,filtros y otras configuraciones de visualización. Además, agrega una pestaña de estilo para personalizar el color, formato de letra y otros elementos visuales del gráfico en construcción.

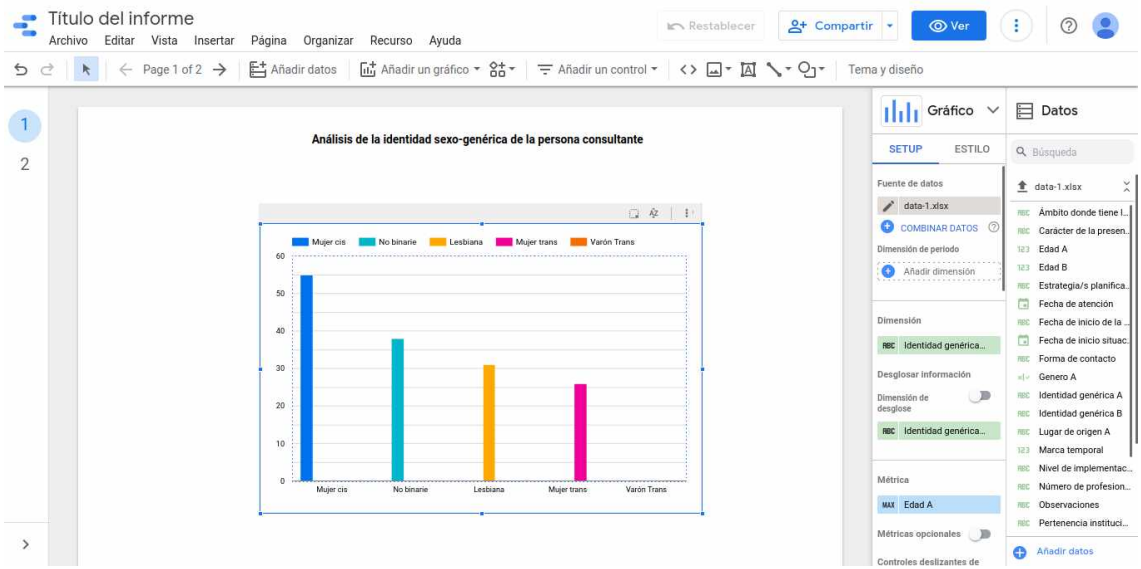


Figura 3.3.4.1 - Creación de visualizaciones

Por último, dado que el objetivo del espacio de trabajo en Data Studio es brindar un informe, también posee herramientas como texto, formas e imágenes que permiten construir un contexto sobre las visualizaciones construidas.

3.3.5 Exportar visualizaciones

Data Studio permite exportar el informe en formato PDF. Además, ofrece la posibilidad de trabajar colaborativamente con otros usuarios, agregándolos al informe o compartiendo el enlace del mismo, así como programar el envío de un mail periódicamente con el informe construido.

3.4 Kibana

Kibana[29] es una interfaz web escalable para la representación visual de datos. Es una aplicación de código abierto bajo la licencia Elastic[30] que forma parte del paquete de productos Elastic Stack (Elasticsearch, Beats, Logstash y Kibana) y que proporciona capacidades de visualización y exploración de datos indexados sobre el motor de analítica Elasticsearch. Fue desarrollado en 2013 en la comunidad de Elasticsearch, y hasta el momento se encuentra disponible en 3 idiomas: inglés, chino y japonés.

Kibana cuenta con una versión básica gratuita y una versión enterprise que se desarrolla en un cloud[31] propio de Elastic u otras infraestructuras en cloud como Amazon Web Services[32] (AWS).

3.4.1 Autenticación

Kibana requiere de una cuenta de usuario para autenticarse en cloud Elastic sobre el que se despliega. Para ello, puede generarse una cuenta o utilizar una existente de Google o Microsoft. Al ingresar en la plataforma, se requiere crear un deploy (despliegue) sobre el que se basará toda la pila de productos, incluido Kibana, y sobre el que luego se podrá trabajar con los distintos productos de la pila Elastic.

---

[29] https://www.elastic.co/es/kibana/
[30] https://www.elastic.co/es/blog/elastic-license-v2
[31] Cloud (nube) es el concepto que refiere al conjunto de servidores accedidos a través de internet, junto con el software y las bases de datos que corren en ellos. Permite a los usuarios acceder a los mismos archivos y aplicaciones desde cualquier dispositivo, ya que se encuentran en centros de datos dispersos en cualquier parte del mundo.
https://www.cloudflare.com/learning/cloud/what-is-the-cloud/
[32] https://aws.amazon.com/

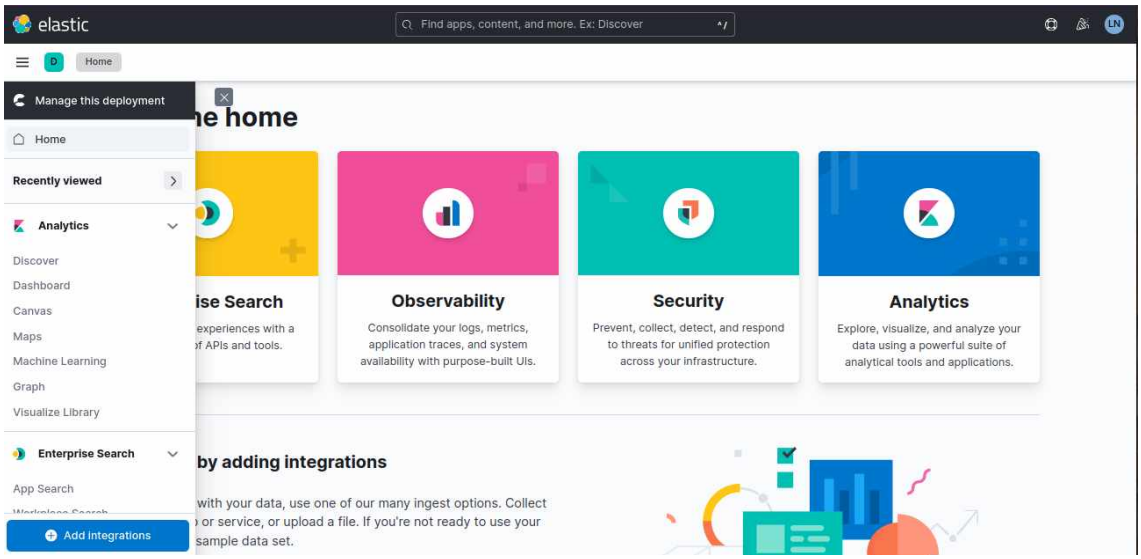


Figura 3.4.1.1 - Ingreso a la plataforma

Como se puede ver en la Figura 3.4.1.1, Kibana es parte de un conjunto de funcionalidades y productos brindados por Elastic.

3.4.2 Carga de datos

Kibana encuentra su potencialidad en la carga de datos a través de la integración de servicios y plataformas, con el objetivo de agregar monitoreo, seguridad, entre otras. También permite cargar datos mediante APIs desarrolladas en JavaScript[33], Java[34], Python[35], Ruby[36] y otros.

Kibana acepta la importación de datos desde archivos de texto delimitados (.csv, .tsv, etc), archivos JSON, delimitados por salto de línea, y archivos de registro (log), con formato común para la marca temporal (Figura 3.4.2.1). Además, ofrece la posibilidad de cargar datos de ejemplo.

[33] https://www.javascript.com/
[34] https://www.java.com/es/
[35] https://www.python.org/
[36] https://www.ruby-lang.org/es/

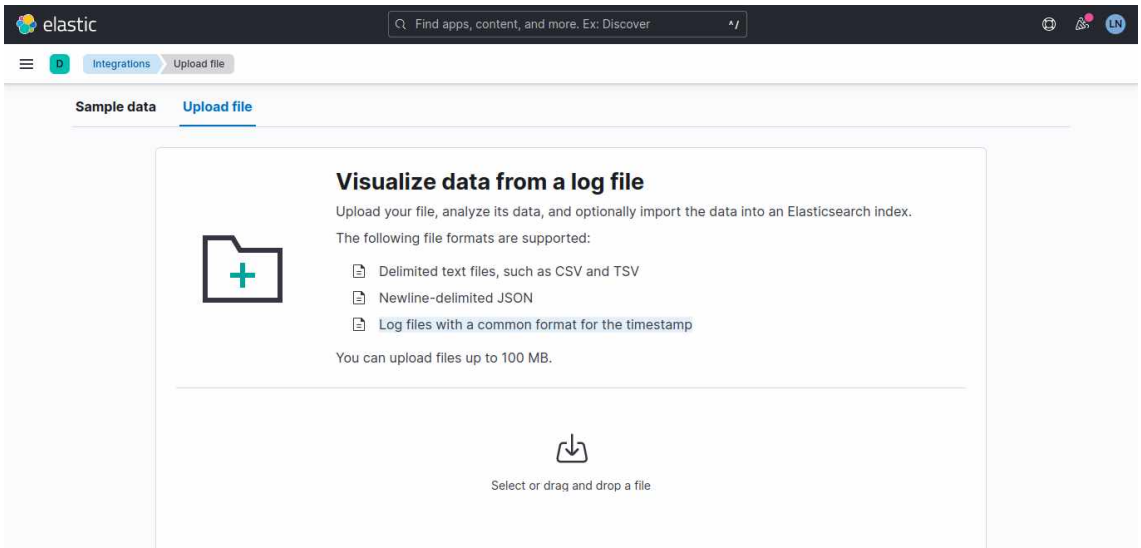


Figura 3.4.2.1 - Cargar datos mediante un archivo

Kibana define por defecto un tamaño máximo de archivo de 100 MB, y permite configurar este valor hasta 1 GB.

3.4.3 Manipulación de datos

Como ya fue mencionado, Kibana está desplegado dentro del motor de Elasticsearch. Dicho motor almacena los datos como documentos JSON, donde cada documento correlaciona un conjunto de claves (nombres de campos o propiedades) con sus valores correspondientes (textos, números, booleanos, fechas, variedades de*valores*, geolocalizaciones u otros tipos de datos). Para manejar estos tipos de documentos, Elasticsearch utiliza una estructura de datos llamada índice invertido, la cual forma una lista de cada palabra única que aparece en cualquier documento e identifica todos los documentos en que ocurre cada palabra. Es decir, cada índice representa una colección de documentos relacionados entre sí. Dicha relación, se genera mediante el proceso de indexación en el cual Elasticsearch almacena documentos y construye un índice invertido.

Kibana requiere crear "*data views*" (vistas de datos) para que el conjunto de datos importado en Elasticsearch sea consumido en la plataforma. Esta creación consiste en un procesos varios pasos (procesar el archivo, crear los índices, crear fuente de

información, cargar los datos y, por último, crear la vista), donde previamente se pueden configurar los tipos de datos para cada campo del conjunto.

3.4.4 Visualizaciones

Kibana, a través de su librería Visualize, permite construir visualizaciones a partir de las fuentes de datos cargadas en la plataforma. Para ello, pone a disposición el listado de campos asociados al conjunto de datos, el listado de tipos de gráficos disponibles y un espacio donde generar las visualizaciones. Mediante el recurso de drag-and-drop, el usuario configura los campos en el espacio de trabajo y la plataforma construye la visualización más aproximada posible para los tipos de datos elegidos (Figura 3.4.4.1). Además, carga un listado de gráficos sugeridos y provee configuraciones (color, etiquetas, entre otras).

Los tipos de visualizaciones que permite construir la plataforma son: tablas, gráficos de barras, gráficos de medición, gráficos de áreas y líneas, mapa de calor, mapa de región y gráficos de proporciones. Una ventaja que presenta Kibana es que calcula automáticamente el tipo de gráfico que mejor se pueda ajustar al campo seleccionado y también, en caso de ser posible, un nuevo campo para acompañarla (por ejemplo, conteo de registros o media del valor).

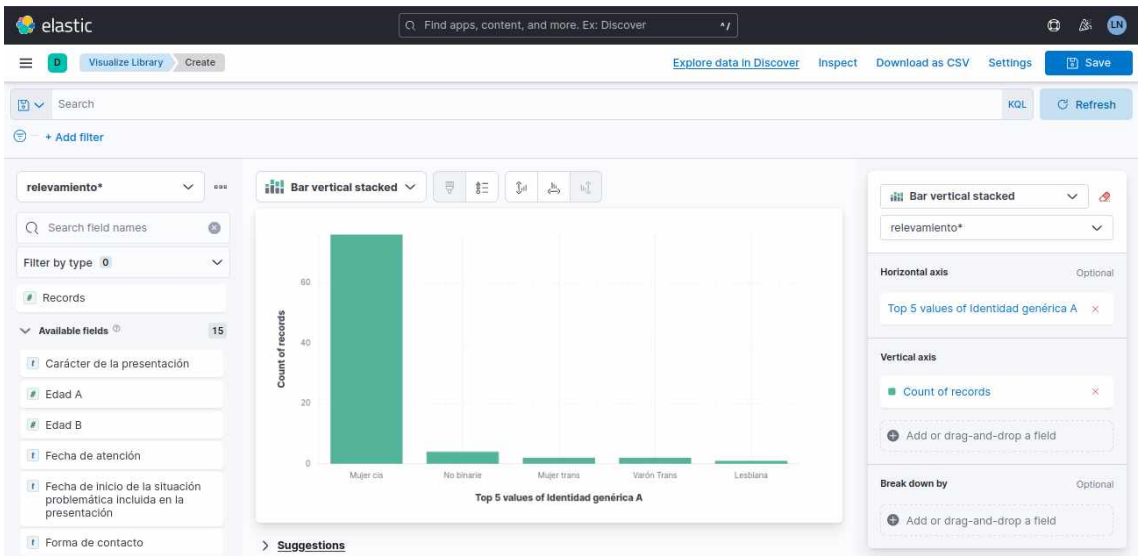


Figura 3.4.4.1 - Creación de una visualización

Por último, Kibana también presenta un apartado para la creación de tableros que combinen varias visualizaciones y otros recursos generados en la plataforma.

3.4.5 Exportar visualizaciones

Kibana no presenta variadas formas de exportar las visualizaciones creadas. Para cada una, ofrece la posibilidad de descargar la información en formato .csv. Luego, para poder generar un reporte a partir de visualizaciones exportadas, sólo es accesible generando una bloque de trabajo en estilo canva y descargándolo en PDF (.pdf).

3.5 Comparación de herramientas

En general, todas las herramientas tienen un proceso de importación de datos mínimo a través de la carga de un archivo siguiendo un formato específico, luego Tableau, Data Studio y Kibana extienden dicho proceso permitiendo conectar con distintas aplicaciones con generación de datos. En cuanto a la manipulación de los mismos, se puede concluir que aún en herramientas básicas como CODAP siempre existe para el usuario una configuración mínima de tipo y nombres. Además, todas comparten el concepto de dimensiones y métricas a la hora de configurar los campos para la construcción de un gráfico, lo cual permite indicar un conjunto común de restricciones y validaciones mínimas sobre la utilización de los datos, aún cuando en términos de complejidad y robustez hay una gran brecha entre las cuatro herramientas. Tableau y Kibana contienen un desarrollo más amplio para el manejo de los datos, permitiendo la manipulación del modelo de datos completo, es decir de toda la estructura que comprende al conjunto de datos importado. En cambio, el desarrollo de CODAP y Data Studio es más acotado, ya que la manipulación de los datos sólo permite hacer un tratamiento individual sobre los campos del conjunto.

Las visualizaciones en todos los casos se van construyendo automáticamente a través de la asignación de dimensiones y métricas, requiriendo al menos una de cada una, salvo en el caso de las tablas y gráficos de valor único (por ejemplo, los sliders). Kibana y Data Studio ofrecen todos los tipos de gráficos habilitados y van ajustando, con los datos existentes, las variables que necesitan para que se pueda construir, en cambio en Tableau y CODAP se restringen los tipos de gráficos habilitados según los tipos de

variables que elige el usuario. La diferencia entre las dos últimas herramientas recae en que Tableau muestra todo el listado de tipos de gráficos y, aún en aquellos deshabilitados, informa cuántas dimensiones y cuántas medidas se requieren para poder utilizar cada uno, sin embargo CODAP muestra el listado de gráficos a medida que se van cargando las variables que habilitan cada tipo, lo cual resulta de mayor dificultad para el usuario al momento de entender cuáles son todos los tipos de gráficos que podría construir.

Por último, en cada una de las plataformas se permite generar y obtener un reporte de los datos trabajados a través de las visualizaciones construidas, lo cual demuestra la necesidad de asignarle un contexto al trabajo realizado, donde los gráficos presentan un medio para unfin, es decir una forma de obtener información y realizar análisis sobre los datos crudos iniciales.

A continuación, se presenta una tabla comparativa donde se resumen los ejes principales para cada herramienta.

Tabla 3.5.1 - Comparación de herramientas

	Tableau	**Codap**	**Data Studio**	**Kibana**
Código abierto		X		
Versión gratuita				
De prueba	X			X
De estudiantes	X			
Completa		X	X	
Disponible en español	X	X	X	
Requiere autenticación	X		X	X
Importación				
Archivo local	X	X	X	X
Conectores a fuentes de datos externas	X		X	X
Manipulación de datos				
A nivel de campo	X	X	X	X
A nivel de modelo	X			X
Visualización				
Automatización de variables			X	X
Tipos de gráficos básicos	X	X	X	X
Tipos de gráficos específicos	X		X	

3.6 Recapitulación

En este capítulo se analizaron cuatro herramientas de visualización y se pudieron obtener las características que construyen el proceso de visualización de datos para el usuario y realizar comparaciones para resolver puntos comunes y diferencias entre las mismas. Los resultados de este análisis de herramientas de visualización influyeron en cómo diseñamos PIBAS. Nos ayudaron a crear una forma precisa de integrar los datos, ofrecer diversos tipos de gráficos y habilitar la creación de tableros e informes.
En el próximo capítulo se pasa a describir la plataforma propuesta.

Capítulo 4: Plataforma PIBAS

PIBAS fue construido como extensión de la herramienta de recolección de datos basada en microservicios llamada DEHIA [Arcidiacono, 2020]. Por esta razón, se distinguirá en el texto qué aspectos se reusan de DEHIA y cuáles se suman a partir de PIBAS.

La plataforma PIBAS está soportada por una arquitectura distribuida, basada en el modelo cliente/servidor donde: el cliente incluye los componentes que resuelven la lógica de presentación al usuario (GUI - Graphical User Interface) y la lógica de negocio del visualizador. El servidor es la herramienta de recolección de datos DEHIA.

En este capítulo se describe la plataforma construida, incluyendo su arquitectura, características principales, funcionalidades y ventajas.

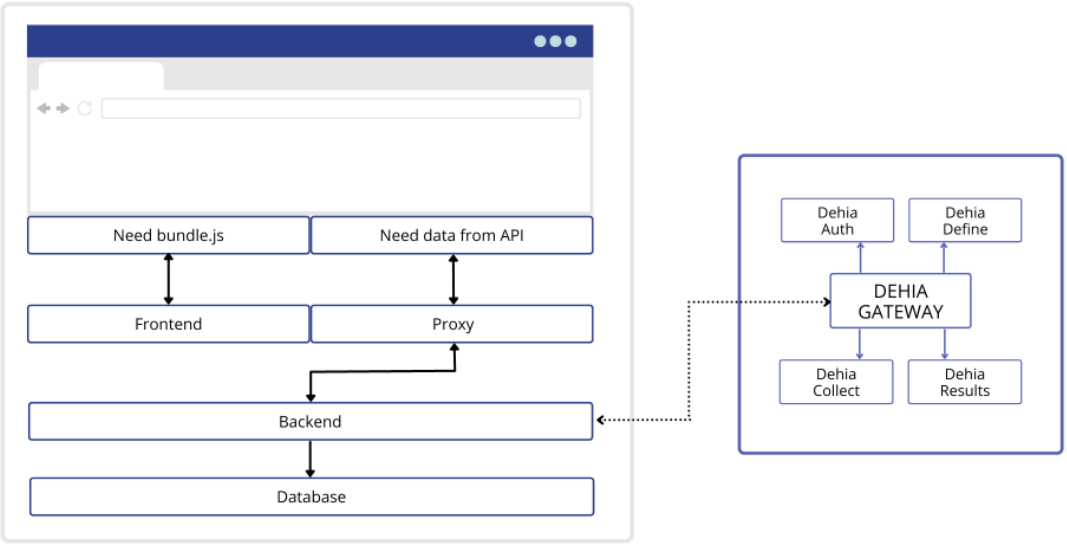


Figura 4.1 - Arquitectura implementada

4.1 Arquitectura Propuesta

La arquitectura propuesta tiene como objetivo proporcionar una solución eficiente y escalable para la presentación de datos a los usuariosfinales. Esta arquitectura consta de dos capas principales: la capa cliente y la capa servidor.

La capa cliente está diseñada para brindar una interfaz de usuario intuitiva y fácil de usar, que permita a los usuarios acceder y visualizar los datos de manera efectiva. Esta capa se compone de un conjunto de componentes de interfaz de usuario que se comunican con la capa servidor a través de una API.

Por otro lado, la capa servidor está constituida por los servicios que ofrece la herramienta de recolección de datos DEHIA, los cuales son consumidos por el componente de visualización. DEHIA actúa como servidor en la arquitectura propuesta, ya que es responsable de gestionar y exponer los datos a través de los servicios que proporciona.

Además, se presentarán alternativas de la arquitectura propuesta, que incluyen enfoques distribuidos e integrados. Estas alternativas se evaluarán en función de sus ventajas y desventajas, y se discutirá (Sección 4.1.4.3) la elección de la arquitectura adecuada para el componente de visualización.

4.1.1 DEHIA

La herramienta de recolección y análisis de datos DEHIA, que se detalla en la Sección 5.3.1, posee una arquitectura distribuida basada en microservicios, los cuales se agrupan acorde a un objetivo específico según las características en común que brindan: autenticación, definición, recolección y resultados. La comunicación se realiza mediante un microservicio intermediario llamado gateway (Figura 4.1), a través del cual la aplicación de recolección de datos y cualquier otra aplicación que lo requiera (PIBAS, en nuestro caso) va a peticionar y consumir los servicios que DEHIA expone.

4.1.2 Capa Cliente

La capa cliente está constituida por las funcionalidades que ofrece la plataforma PIBAS. En particular, se cubren los siguientes aspectos:

1. Listado de actividades creadas en DEHIA por la persona usuaria logueada, junto con sus tareas asociadas.

2. Administración de visualizaciones para una actividad (Figura 4.1.2.1), que incluye las opciones de listar, crear, eliminar, descargar, obtener el detalle de cada una, con su valor al día de hoy y su histórico, y seleccionar como favorita.
3. Administración de tableros que agrupan visualizaciones (Figura 4.2.1.2), crear, editar, eliminar, descargar y seleccionar como favorito.
4. Administración de un panel de control del usuario, que incluye un resumen estadístico de las actividades creadas en DEHIA, gráficos y tableros favoritos.

La estructura visual de la plataforma tiene como base un layout en forma de L invertida, que contiene el menú de secciones disponibles, la información del usuario y todas aquellas características generales. Además, la pantalla de inicio, los listados, visualizaciones y elementos propios de cada sección se presentan en el centro de la pantalla y, particularmente, en forma de cards que organiza y agrupa las features (características) con las que puede trabajar el usuario.

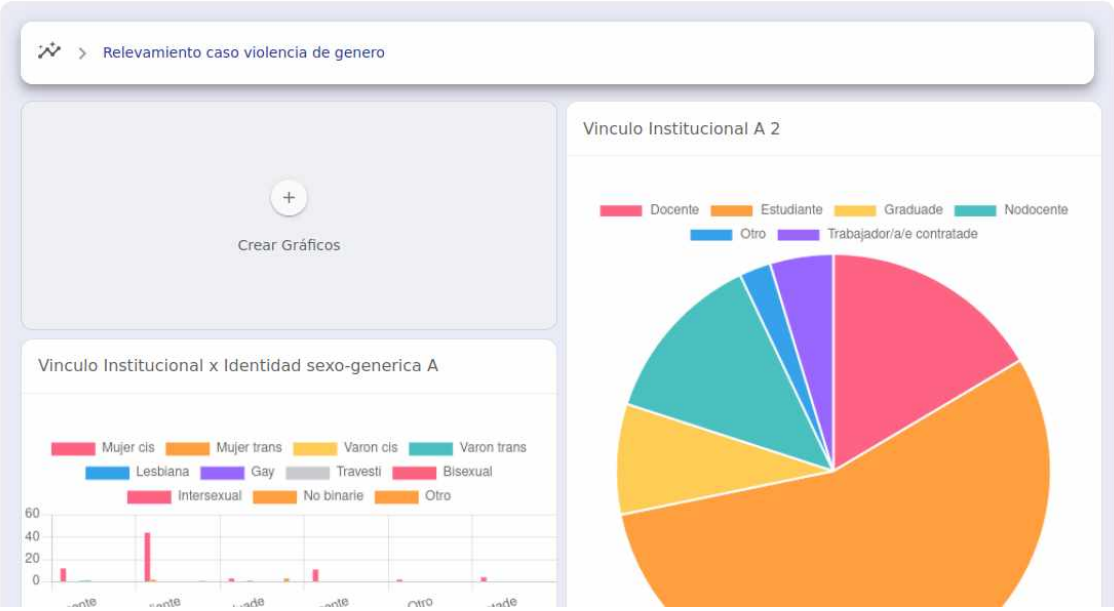


Figura 4.1.2.1 - Administración de gráficos por actividad

La generación de visualizaciones está basada en los parámetros, llamados indicadores, que configure la persona usuaria. El tipo de gráfico seleccionado es el que permitirá acceder a los campos involucrados en la visualización y las medidas requeridas para la

construcción del mismo. Estas últimas pueden depender de la categoría de la visualización, sin embargo se referirán siempre como columnas yfilas a modo de mantener una experiencia de usuario amigable y consistente al resto de las herramientas de visualización conocidas.

Además, a través de la creación de tableros, se agrega la posibilidad de agrupar las visualizaciones construidas acorde a las necesidades de las personas usuarias.

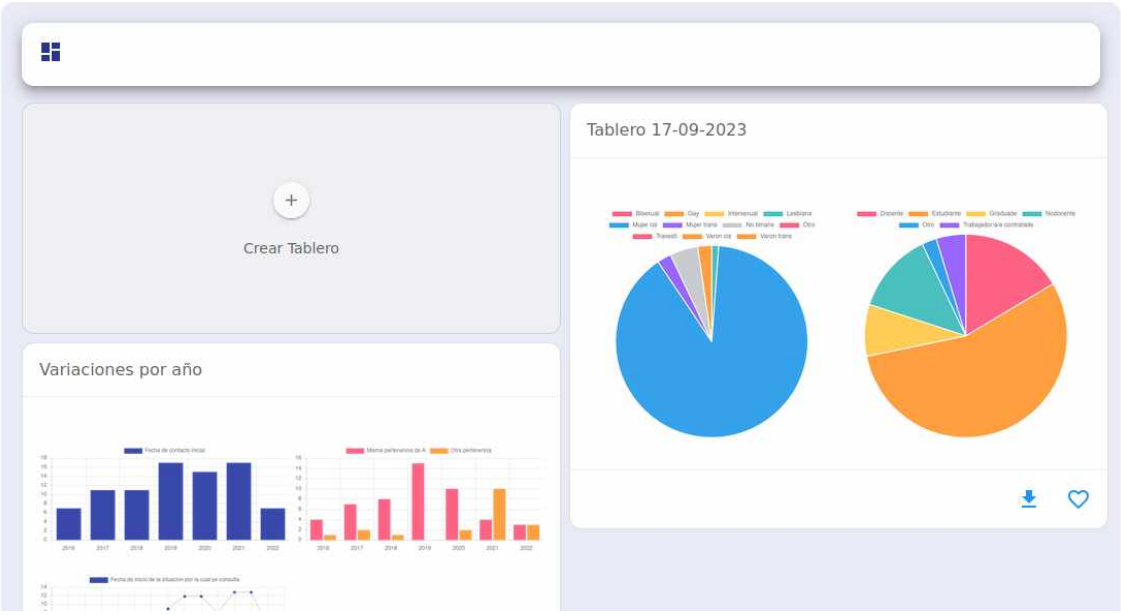


Figura 4.2.1.2 - Administración de tableros

Esta capa, integrada en la arquitectura propuesta, forma la base de la interacción entre los usuarios y los datos, impulsando la toma de decisiones informadas y la optimización de las operaciones.

4.1.3 Capa Servidor

El servidor es la herramienta de recolección de datos DEHIA. La capa servidor comprende a los servicios que expone la herramienta de recolección de datos y consume el visualizador. En este caso, el componente de visualización requiere de los siguientes servicios:

- Servicio de autenticación: este servicio tiene como objetivo validar la identidad de los usuarios que ingresan a la aplicación. Para ello, la API se encarga de identificar a los usuarios mediante un token de identificación de Google enviado desde la aplicación web. Algunas consultas como los listados de atributos y actividades públicas son accesibles sin necesidad de autenticarse. Mientras que, para las acciones de consulta de resultados, u otras consultas restringidas, son accesibles sólo para los autores registrados.
- Servicio de definición: este servicio permite obtener el listado de actividades de la persona usuaria junto con sus tareas asociadas.
- Servicio de resultados: este servicio permite consultar sobre los resultados registrados. Mediante el código de una actividad, brinda el listado de "entradas" que se corresponden con cada instancia de respuesta. Cada entrada consiste en un listado de pares (nombre de tarea, respuesta).
- Gateway: este servicio actúa como intermediario entre los servicios de la herramienta de recolección de datos y el visualizador. En particular, expone una API que permite consultar resultados mediante el uso del servicio de resultados antes mencionado.

4.1.4 Alternativas de la arquitectura

Un enfoque para resolver este modelo consiste en tener definida cada parte (cliente y servidor) alojadas por separado (diferentes dominios). Dicho enfoque se presenta como una ventaja en softwares que requieren de un servidor robusto que pueda funcionar como una API completamente desacoplada del cliente que lo consume.

Sin embargo, otro enfoque posible incluye tener los servicios integrados al cliente, siendo alojados en el mismo servidor (bajo el mismo dominio). Este enfoque requiere que los accesos al servidor sean limitados, es decir que la lógica principal de la aplicación esté concentrada en el cliente.

4.1.4.1 Enfoque distribuido

La arquitectura distribuida se refiere a una estructura de software en la que los diferentes componentes de la aplicación se ejecutan en diferentes nodos de red. Los

nodos pueden utilizar diferentes plataformas o tecnologías. Este enfoque tiene como ventaja principal la escalabilidad y la posibilidad de distribuir la carga de trabajo en múltiples servidores. Sin embargo, también puede tener desventajas en términos de complejidad de diseño, implementación y gestión, así como en la seguridad y el rendimiento de la aplicación.

En este enfoque, se separan totalmente el frontend y el backend (Figura 4.1.4.1.1), utilizando un servidor para los recursos del frontend y otro para la información del sistema. Además, se utilizan diferentes dominios para cada uno. Este enfoque es especialmente útil cuando se tienen backends pesados que solo se utilizan como servidores API a los que se puede acceder en todo momento sin importar el contexto.

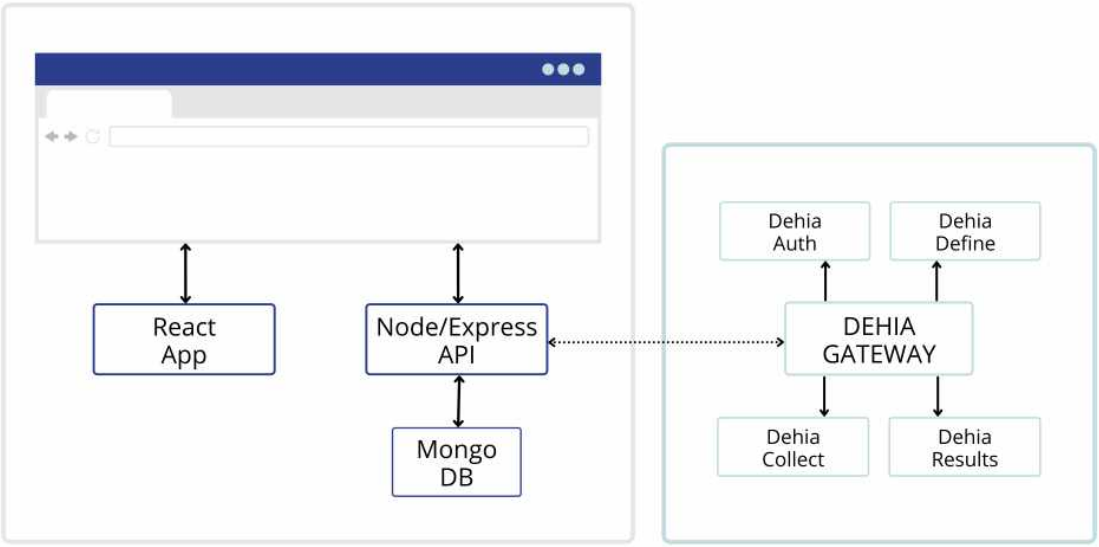


Figura 4.1.4.1.1 - Enfoque distribuido

4.1.4.2 Enfoque integrado

La arquitectura integrada, también conocida como monolítica, se refiere a una estructura de software en la que todos los componentes y funcionalidades de la aplicación están integrados en una sola unidad (Figura 4.1.4.2.2). Esta arquitectura resulta adecuada en situaciones donde se necesita una mayor simplicidad en el diseño y la implementación, menor complejidad en la gestión de datos y recursos, y mayor facilidad para realizar pruebas y depuración.

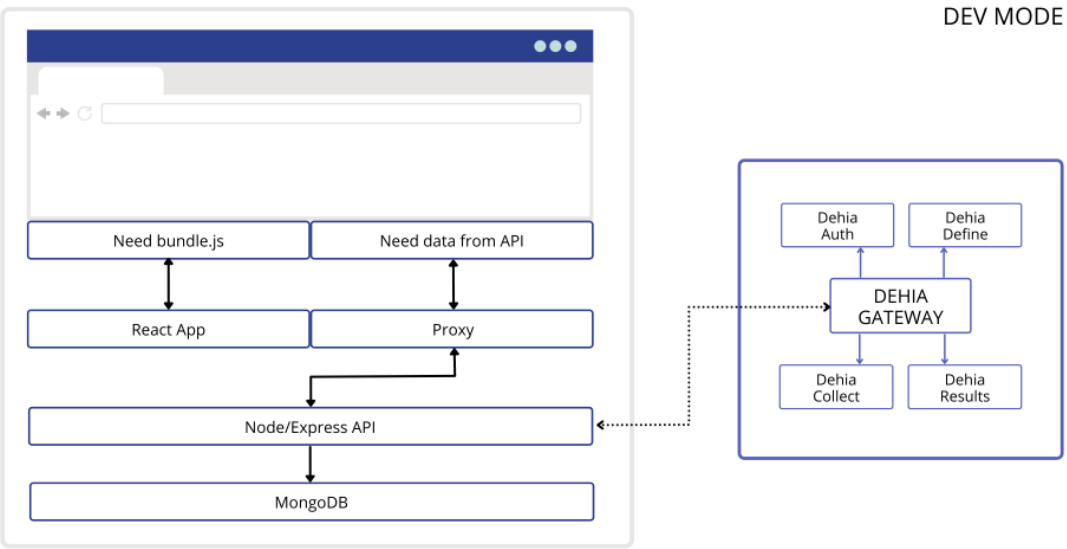


Figura 4.1.4.2.1- Enfoque integrado en desarrollo

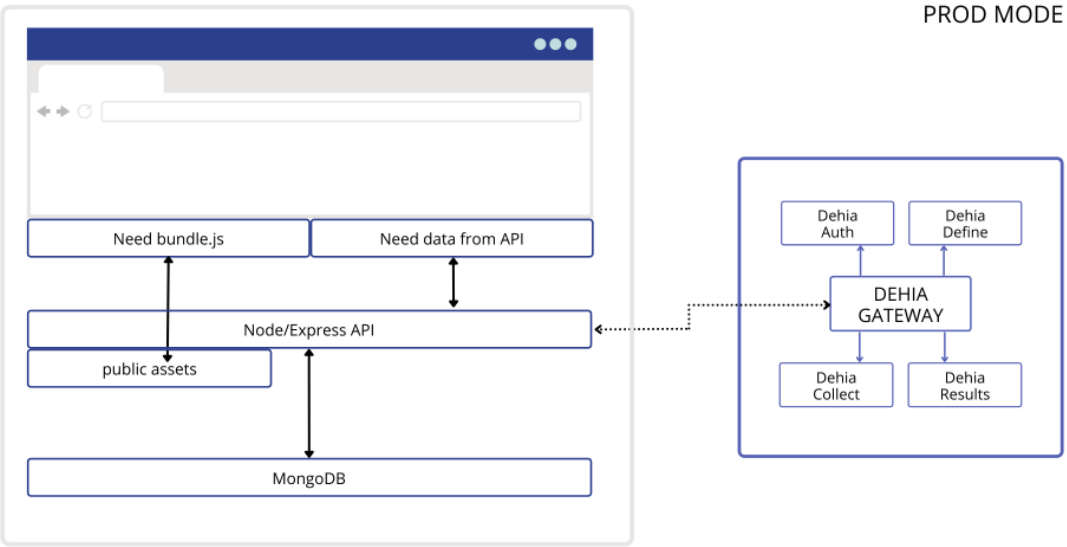


Figura 4.1.4.2.2 - Enfoque integrado en producción

4.1.4.3 Elección de la arquitectura

El uso de una arquitectura distribuida puede ser más complejo que una arquitectura integrada debido a la necesidad de coordinar múltiples componentes en diferentes ubicaciones y plataformas. También, es importante tener en cuenta que la comunicación entre los componentes en una arquitectura distribuida se realiza a través de la red, lo que puede influir en el rendimiento y en la seguridad de la aplicación [Tanenbaum & Van Steen, 2017].

Siendo que el objetivo de la plataforma PIBAS es dar solución a un componente de visualización, del lado del cliente de la aplicación, se resolvió utilizar el segundo enfoque mencionado donde la lógica principal del negocio se implementa del lado del cliente y el servidor complementa a la aplicación como conector hacia la herramienta de recolección de datos, en este caso DEHIA.

Para ello, se implementa una arquitectura integrada en la cual el backend está conectado con el frontend a través de un proxy (Figura 4.1.4.2.1). En este caso, los accesos al backend deben estar restringidos y la lógica principal debe encontrarse en el frontend, por lo cual se considera adecuado para el contexto de la plataforma PIBAS y su comunicación con DEHIA, o cualquier otra herramienta de recolección de datos de la cual deba servirse.

Elegimos la arquitectura integrada debido a las ventajas que ofrece en términos de simplicidad, facilidad de gestión y menor complejidad en la implementación y las pruebas. Además, dado que el enfoque de la aplicación se centra principalmente en la visualización y el análisis de datos para usuariosfinales, no requerimos manejar cargas de trabajo intensivas.

A continuación, detallamos algunas de las ventajas que ofrece la arquitectura elegida:

- Simplicidad en el diseño y la implementación: cuando se integran todas las funcionalidades en una sola unidad, el diseño y la implementación se vuelven más simples y menos propensos a errores. Además, los desarrolladores pueden trabajar en el código de una manera más eficiente, ya que no tienen que preocuparse por la complejidad que tiene una arquitectura distribuida.

Según Andrew S. Tanenbaum y Maarten van Steen en el libro "Distributed Systems: Principles and Paradigms", la simplicidad es una de las principales ventajas de una arquitectura integrada. Los autores argumentan que una arquitectura distribuida puede ser más difícil de diseñar, implementar y mantener debido a la complejidad de la comunicación entre los diferentes componentes.

- Facilidad para realizar pruebas y depuración: en una arquitectura integrada, las pruebas son más sencillas ya que todas las funcionalidades están en una sola unidad. Esto significa que los desarrolladores pueden probar la aplicación de manera más eficiente y solucionar errores de forma más rápida.

En el libro "Clean Architecture: A Craftsman's Guide to Software Structure and Design", Robert C. Martin se destaca lo importante de realizar pruebas en el proceso del desarrollo de software. Según Martin, tienen que ser una parte integral del proceso de diseño de la arquitectura, y una arquitectura integrada facilita la realización de pruebas y la detección de errores.

- Menor complejidad en la gestión de datos y recursos: en una arquitectura integrada, la gestión de los datos y los recursos es más simple, ya que todas las funcionalidades están en una sola unidad. Esto significa que los desarrolladores no tienen que preocuparse por la complejidad de la comunicación entre los diferentes componentes y la gestión de recursos en una arquitectura distribuida.

En el libro "Software Architecture in Practice", Len Bass, Paul Clements y Rick Kazman se explica cómo una arquitectura integrada simplifica la gestión de recursos. Los autores argumentan que la gestión de recursos en una arquitectura distribuida puede ser más compleja y requiere de una planificación cuidadosa, mientras que una arquitectura integrada puede simplificar esta tarea.

4.2 Implementación PIBAS

En esta sección, se describe la implementación de la plataforma PIBAS. Se detallan las tecnologías utilizadas tanto en el frontend como en el backend, así como la metodología de desarrollo empleada. Además, se proporciona una visión general del proceso de implementación y se destacan las características principales de la plataforma.

4.2.1 Integración de datos

La recolección de datos es solo el primer paso en la obtención de información valiosa. Para que los datos sean útiles, es necesario procesarlos y transformarlos en información normalizada. En esta sección, se explicará cómo se llevó a cabo la limpieza y la integración de los datos recolectados para su posterior visualización.

4.2.1.1 Limpieza de datos

Los datos recolectados en ocasiones presentan diversos errores y fallas que dificultan su procesamiento y análisis. Para la implementación de PIBAS se pudieron conseguir datos reales que provenían de un formulario de Google, en formato de archivo excel. Siendo dicho formulario de un formato bien definido y teniendo además la gran mayoría de preguntas con un formato de respuesta cerrado (opción predefinida), se esperaban que los datos reales sean limpios y normalizados, en su gran mayoría, y sin embargo no fue este el caso. Al analizar los datos se vio que existían respuestas idénticas con formatos distintos, campos obligatorios con una respuesta vacía y respuestas que a priori eran distintas perofinalmente se englobaban dentro de la misma opción de respuesta.

A partir de los errores identificados, la limpieza de datos incluyó la eliminación de valores atípicos, la corrección de errores de formato, la normalización de datos y la eliminación de datos incompletos. En este caso, al tener pocos registros y no ser el objetivo de esta tesina hacer un procesamiento exhaustivo de los datos crudos, el proceso de limpieza se realizó manualmente.

Uno de los principales desafíos fue la normalización de los datos ya que, como se mencionó, para un mismo valor de respuesta existían distintas versiones, donde algunas estaban en mayúsculas, otras en minúsculas, y otras utilizaban diferentes abreviaciones y variaciones de las palabras. Para abordar este problema, se decidió normalizar todas las respuestas en minúsculas y unificar las diferentes variaciones de una misma palabra.

ARTES
ARTES
BACHILLERATO DE ARTES
Facultad de Artes
Facultad de Artes

Figura 4.2.1.1.1 - Subconjunto de resultados “UDA/Área interviniente” sin normalizar

Violencias de género	Identidad de género
Violencia de género	Discriminación por identidad de género

Figura 4.2.1.1.2 - Subconjunto de resultados “Temática” sin normalizar

Es importante destacar que la limpieza de datos es un paso crítico en el proceso de integración de datos, ya que garantiza que los datos estén completos, precisos y coherentes. La limpieza de datos también ayuda a reducir el riesgo de errores y a garantizar que los resultados sean confiables y precisos. Si se utilizan datos no limpios para hacer visualizaciones de datos, es probable que se generen visualizaciones erróneas que pueden afectar la toma de decisiones.

Después de la limpieza de datos, se generó un archivo .csv para poder procesar los datos en la herramienta de recolección de datos.

4.2.1.2 Integración de datos a DEHIA

En el proceso de integración de datos a DEHIA, se consideraron dos escenarios posibles:

- Uso de la plataforma DEHIA interactuando directamente a través de su interfaz.
- Utilización de la API de DEHIA, con un enfoque más técnico y especializado.

Teniendo en cuenta estos escenarios elegimos utilizar la API de DEHIA debido a su enfoque preciso y eficiente. A través de la API, pudimos automatizar el proceso de integración y transferencia de datos de manera más confiable y rápida, minimizando la posibilidad de errores en la entrada de datos.

Para la integración de los datos del caso de estudio con la herramienta DEHIA, se desarrolló un script en Node.js que hacía uso del endpoint /entries de la API de la herramienta.

El script recibe como entrada el archivo .csv generado en la limpieza de datos (Sección 4.2.1.1) y lo procesa para enviar cada registro de manera individual a la herramienta, de manera que los datos queden registrados por DEHIA, y sean consumidos posteriormente por PIBAS.

El proceso de integración fue exitoso y todos los datos fueron almacenados en la herramienta de recolección de datos DEHIA.

4.2.2 Diseño de interfaz de usuario

El diseño de la interfaz de usuario es uno de los aspectos más importantes en el desarrollo de cualquier plataforma, ya que influye directamente en la experiencia del usuario. Como soporte para desarrollar el diseño, en PIBAS se optó por utilizar la herramienta de diseño Canva para crear las pantallas necesarias para cumplir con las funcionalidades de la herramienta.

El diseño se enfocó en hacer que la experiencia del usuario sea lo más efectiva y satisfactoria posible con un enfoque en la organización y el diseño de cada pantalla para permitir a los usuarios visualizar y analizar datos de manera eficiente, independientemente de su nivel de experiencia o conocimiento técnico.

La navegación entre las funcionalidades se encuentra contenida dentro del mismo espacio de la pantalla (Figura 4.2.2.1).

Figura 4.2.2.1 - Inicio

Esta forma minimalista de organizar la aplicación, junto con la agrupación en tarjetas y el menú de las principales funcionalidades, permite que las personas que utilicen la plataforma encuentren la complejidad sólo en el procesamiento de las visualizaciones y no en la interacción con la plataforma. Las decisiones de diseño UX/UI se desprenden de la forma de interacción más utilizada actualmente en el mercado, implementada en base a un template gratuito que provee la librería de componentes visuales llamada Material UI. Además, dicha librería es la elegida para el desarrollo de los componentes utilizados en la plataforma ya que provee gran parte de los elementos básicos necesarios para la interfaz de usuario de una aplicación web.

La pantalla de inicio de sesión (Figura 4.2.2.2) se diseñó para permitir a las personas usuarias iniciar sesión de manera rápida y sencilla, utilizando su cuenta de Google.

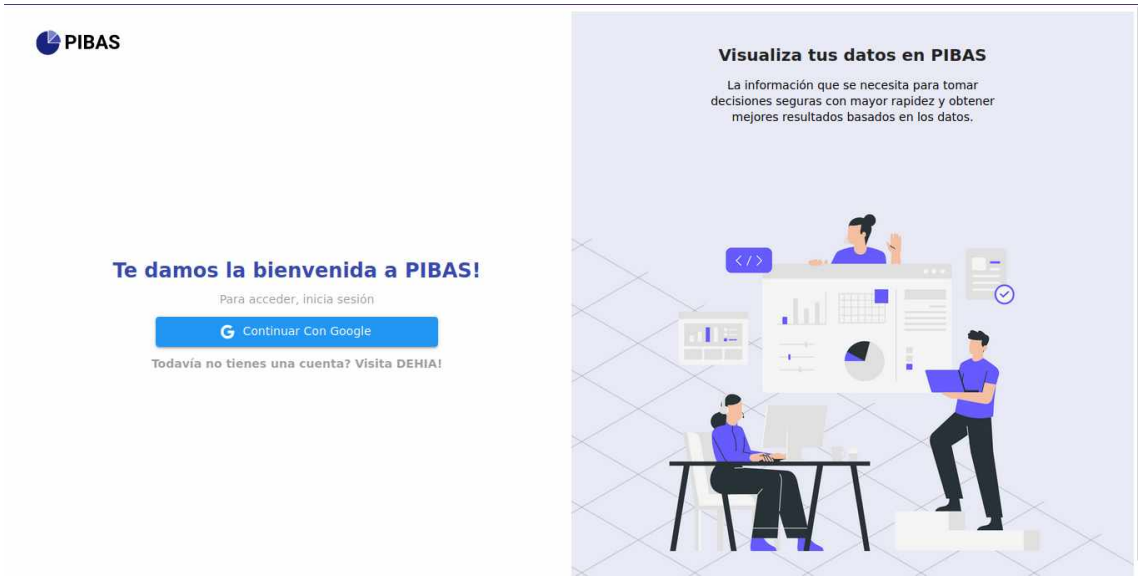


Figura 4.2.2.2 - Inicio de sesión

La pantalla de listado de actividades (Figura 4.2.2.3) es donde los usuarios visualizan las actividades definidas previamente en DEHIA, junto con sus tareas asociadas. Desde esta pantalla, las personas usuarias tienen acceso a los gráficos asociados a cada actividad con tan sólo un botón, facilitando la navegación y construyendo al mismo tiempo una concordancia en elflujo natural de la relación conceptual que tienen las características principales de la aplicación.

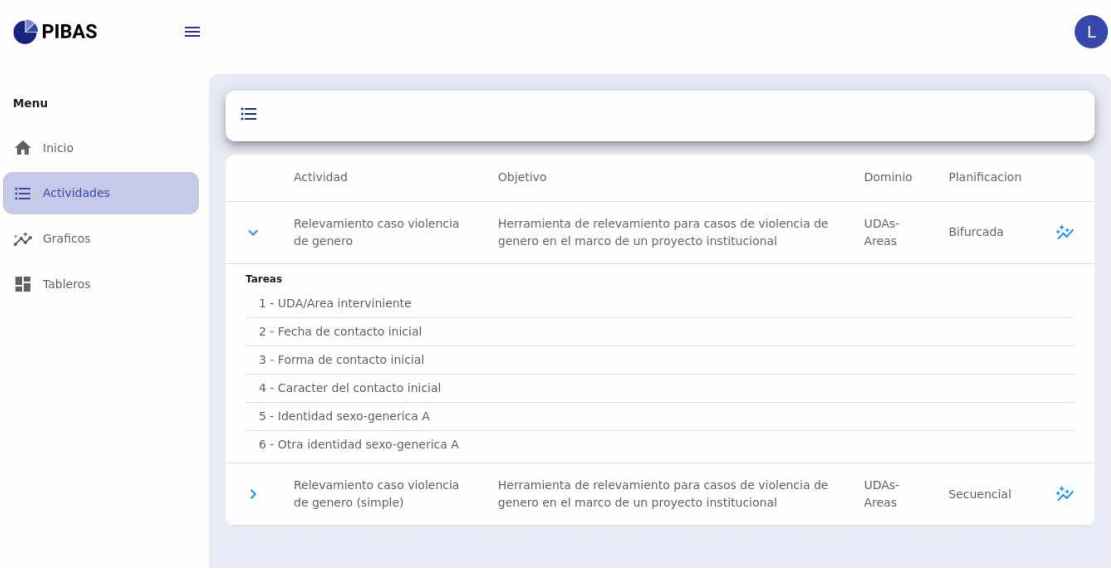


Figura 4.2.2.3 - Listado de actividades

La pantalla de generación de gráficos (Figura 4.2.2.4) permite a los usuarios visualizar acceder a todos los indicadores para crear gráficos de manera rápida y fácil. Las visualizaciones pueden ser creadas en la misma pantalla a través de pestañas, lo cual brinda potencialidad yflexibilidad a la hora de construir gráficos sobre un mismo conjunto de datos.

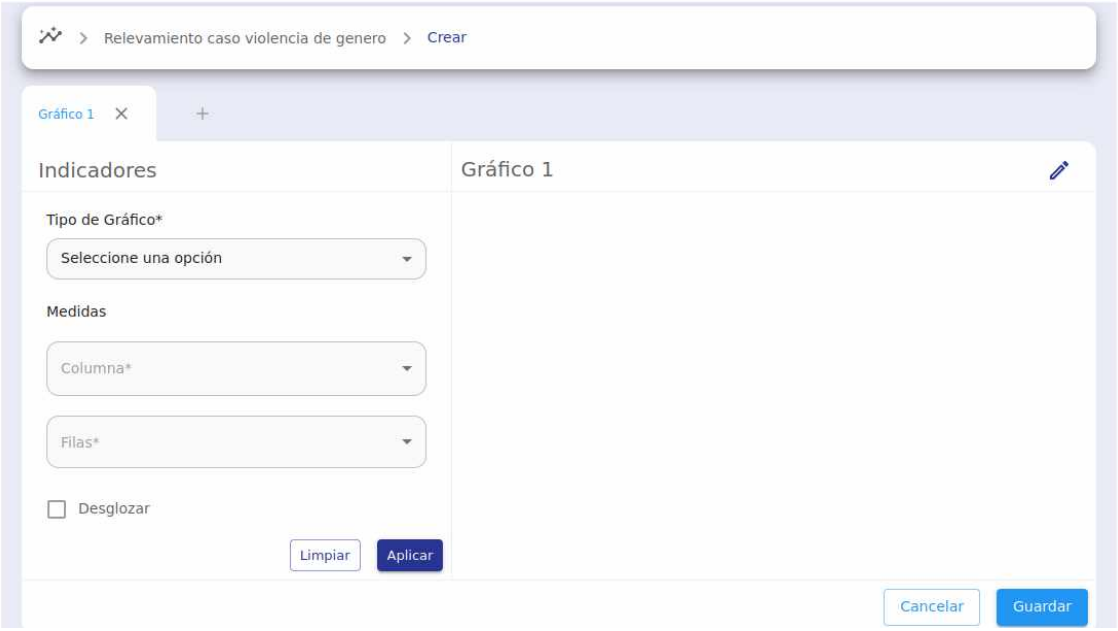


Figura 4.2.2.4 - Creación de gráficos

Por otra parte, en la pantalla de creación de un tablero se trabajó con el recurso de *drag-and-drop*(arrastra y suelta) mediante el cual se pueden generar conjuntos de gráficos trasladando a los mismos con un movimiento similar al de un trabajo manual como el de llenar una caja, agarrando los elementos a través del mouse e introduciéndolos en el contenedor que representa al tablero.

Figura 4.2.2.5 - Creación de tablero

La elección de colores en el diseño de una plataforma web es importante porque puede influir en la experiencia del usuario y en la percepción de la marca. Para PIBAS, se seleccionó una paleta de colores violetas azulados heredados de un enfoque transversal hacia una visualización de datos con perspectiva de género.

4.2.3 Autenticación

La autenticación es un elemento crucial en la seguridad de cualquier aplicación web, ya que permite asegurarse de que solo los usuarios autorizados tengan acceso a las funcionalidades del sistema.

En el marco de esta tesis, se han implementado dos tipos de autenticación para garantizar la seguridad de la herramienta que se está desarrollando. Por un lado, se ha implementado una autenticación basada en sesiones para la comunicación entre el frontend en React y el backend en Node.js, y por otro lado, se ha implementado una autenticación basada en tokens JWT para la comunicación entre el backend y la herramienta de recolección de datos DEHIA.

4.2.3.1 Autenticación basada en sesiones

La autenticación basada en sesiones se utiliza para validar la identidad del usuario en cada solicitud que se realiza desde el frontend al backend. En este caso, se implementó la autenticación basada en sesiones utilizando cookies. Cuando el usuario inicia sesión en la aplicación, se crea una sesión en el servidor backend que se almacena en una cookie en el navegador del usuario. La cookie se envía con cada solicitud al servidor backend y se utiliza para autenticar al usuario.

Para implementar la autenticación basada en sesiones en el backend de la aplicación que se está desarrollando con Node.js, se utilizó una biblioteca de autenticación de sesión llamada express-session. Esta biblioteca proporciona una forma sencilla de crear y administrar sesiones de usuario en una aplicación Node.js.

En la configuración de express-session, se estableció una cookie de sesión segura con una vida útil definida. Esto garantiza que la sesión del usuario expire después de un período de inactividad definido. Además, se estableció un secreto compartido para la firma de la cookie de sesión, lo que agrega una capa adicional de seguridad. También, se utiliza MongoDB para almacenar las sesiones de usuario en la base de datos. Esto proporciona una forma segura y escalable de almacenar las sesiones de usuario y garantiza que los usuarios puedan mantener su sesión activa incluso después de reiniciar la aplicación.

4.2.3.2 Autenticación basada en tokens

La autenticación basada en tokens JWT (JSON Web Token) es un mecanismo de autenticación que se utiliza para proteger las rutas de acceso y los recursos de una aplicación web. En este mecanismo, cuando un usuario se autentica, el servidor genera un token JWT que se devuelve al cliente y se almacena en su navegador o aplicación móvil. Luego, cada vez que el cliente realiza una solicitud, se envía el token JWT en la cabecera de la petición, y el servidor verifica la validez del token antes de procesar la solicitud.

El token JWT consta de tres partes: la cabecera, el cuerpo y lafirma. La cabecera contiene información sobre el tipo de token y el algoritmo de encriptación utilizado parafirmar el token. El cuerpo contiene información adicional sobre el usuario, como su nombre de usuario o rol, y puede personalizarse según las necesidades de la aplicación. Lafirma se utiliza para garantizar la integridad del token y asegurarse de que no ha sido manipulado.

En la implementación de la autenticación para la comunicación entre el backend Node.js y la herramienta de recolección de datos DEHIA, se utiliza la autenticación basada en tokens JWT. Esto garantiza una mayor seguridad en el intercambio de datos entre ambas aplicaciones.

Para lograr la autenticación basada en tokens JWT, el backend Node.js se encarga de realizar la petición del token a la API de la herramienta DEHIA. Al hacer una solicitud de autenticación a la herramienta DEHIA, se recibe un access token que se guarda en una variable de sesión y se utiliza en todas las solicitudes posteriores a la herramienta.

La implementación de la autenticación basada en tokens JWT garantiza la seguridad en el intercambio de información entre el backend Node.js y la herramienta DEHIA. Esto se logra gracias a que el token JWT contiene toda la información necesaria para validar la identidad de quien realiza la solicitud y que la información que se intercambia es la correcta.

4.2.4 Tecnologías Utilizadas

En la implementación de la plataforma PIBAS se utilizaron diferentes tecnologías para el desarrollo tanto del frontend como del backend. En esta sección se describirán dos de las tecnologías utilizadas en el proyecto.

4.2.4.1 Frontend: React.js

React.js es una biblioteca de JavaScript de código abierto utilizada para la creación de interfaces de usuario. Fue desarrollada por Facebook y se utiliza para crear componentes de interfaz de usuario reutilizables. En PIBAS, se utilizó React.js para la

creación de la interfaz de usuario de la plataforma. Entre las ventajas de utilizar React.js se encuentran la facilidad para crear componentes reutilizables, la posibilidad de trabajar con una arquitectura de componentes y la capacidad de manejar grandes cantidades de datos sin comprometer el rendimiento.

4.2.4.2 Backend: Node.JS con Express JS

Node.js es un entorno de tiempo de ejecución de JavaScript que se utiliza para crear aplicaciones del lado del servidor. En PIBAS, se utilizó Node.js junto con el framework Express JS para la creación del backend de la plataforma que se utiliza para crear aplicaciones web y APIs. Entre las ventajas de utilizar Node.js y Express JS se encuentran la facilidad para crear servidores web, la capacidad de trabajar con múltiples solicitudes simultáneamente y la posibilidad de integrar diferentes bibliotecas y módulos de terceros.

4.3 Recapitulación

En este capítulo se describió la arquitectura específica del prototipo desarrollado, para luego detallar la funcionalidad de cada componente y las tecnologías utilizadas. En el próximo capítulo se tratará el caso de estudio en el cual se aplicó este prototipo.

Capítulo 5: Caso de estudio

Una de las principales motivaciones a la hora de planificar la solución tecnológica era que fuese aplicable dentro de la esfera de los derechos de las mujeres y diversidades, las desigualdades de género y toda forma de desigualdad y violencia. Es por esto que la aplicación de la resolución a una herramienta de relevamiento para casos de violencia de género se presentó como una oportunidad para mostrar un análisis de datos que brinde información cualitativa y que además contribuya, a través de visualizaciones enriquecedoras, a construir un ámbito más justo e igualitario en nuestra Universidad Nacional de La Plata.

5.1 Problemática a resolver

El caso de estudio elegido para mostrar el potencial de PIBAS es un proyecto institucional de fortalecimiento de protocolos y de estrategias de intervención sobre situaciones que configuran violencia de género en el ámbito universitario, particularmente en la UNLP, coordinado por la Dirección de Género, Diversidad y Derechos Humanos con las Unidades de Atención (UDAS) y Áreas de Género, Diversidad y Derechos Humanos existentes en las unidades académicas de la Universidad.

Dicho proyecto institucional utiliza un formulario de Google (Figura 5.1.1) para registrar los datos de cada situación que configura violencia por razones de género en el ámbito universitario, donde cada responsable de las distintas UDAS es la persona encargada de completar la información. De esta manera, el proyecto busca recolectar información que le permita tomar decisiones y generar políticas universitarias en pos de mejorar los protocolos de intervención y cuidados.

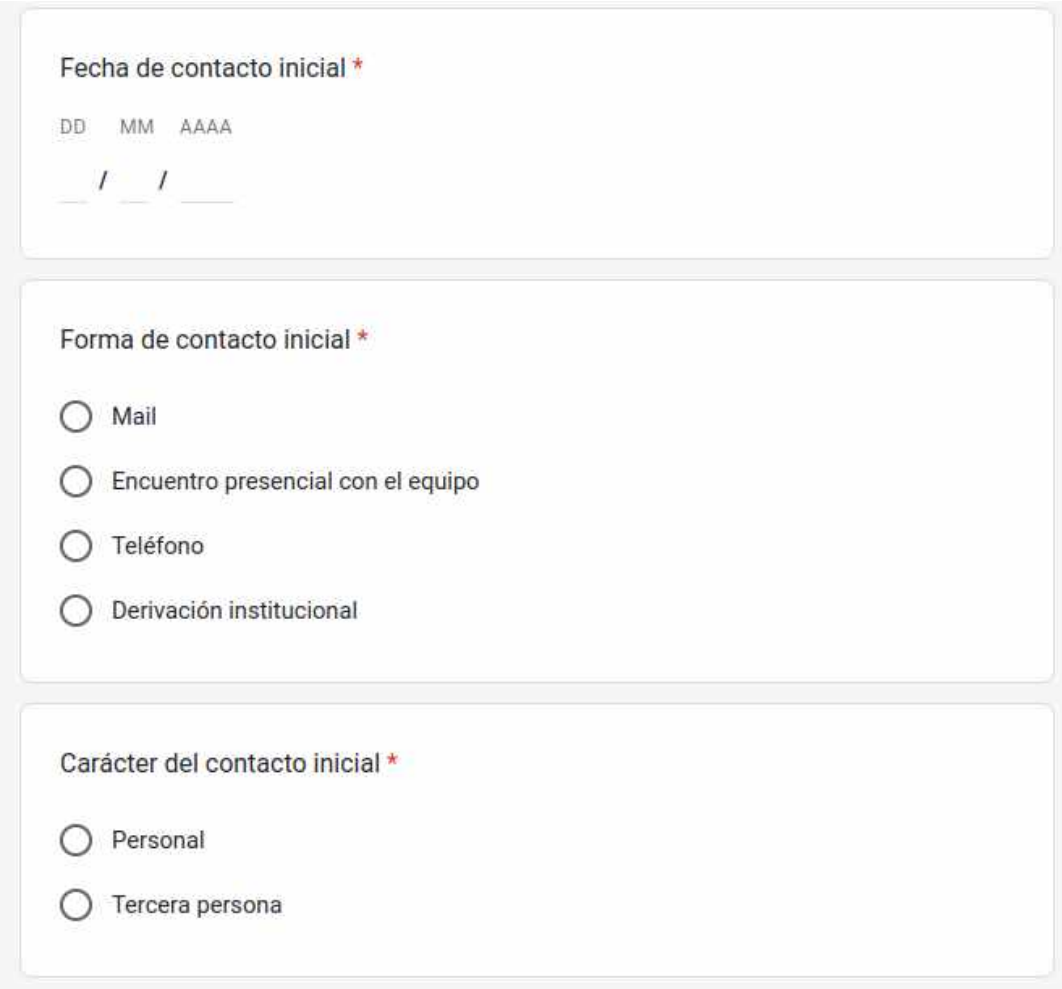


Figura 5.1.1 - Formulario de Google

La problemática a resolver radica en la necesidad de fortalecer laflexibilidad y potencialidad de las visualizaciones que se pueden obtener a partir de los datos recolectados, para su posterior análisis. En consecuencia, aunque la herramienta de formularios de Google provee ciertas visualizaciones predefinidas que permitirían un análisis básico, es deseable que el proyecto universitario pueda contar con visualizaciones personalizadas que permita una mayor transversalidad en los indicadores y en el potencial análisis que pueda brindar cada una.

En este contexto, la solución planteada con el uso de PIBAS permitirá una visualización más clara y detallada de los datos recolectados, lo que a su vez facilitará la identificación de patrones y tendencias en la violencia de género en la Universidad. Además, PIBAS permitirá un análisis más profundo y cualitativo de los datos, lo que

contribuirá a mejorar la evaluación y el seguimiento de la política institucional de género y, en última instancia, a promover un ambiente más justo e igualitario en la Universidad.

5.2 Solución existente

La solución actual hace el relevamiento de datos con una hoja de cálculos (Figura 5.2.1) con los datos recolectados hasta el momento, donde se identificó también que este recurso es en sí mismo utilizado como forma de relevamiento, completando los datos directamente en la hoja de cálculos. A su vez, a partir de los resultados obtenidos, la herramienta formulario de Google provee ciertos tipos de visualizaciones predefinidas que toman como indicador el porcentaje por respuesta para cada pregunta; hay que tener en cuenta que estas visualizaciones generadas automáticamente sólo consideran los datos generados a través del formulario, por lo que todos aquellos casos que se cargan manualmente en la hoja de cálculos no serán procesados por la herramienta de Google.

	B	C	D	E	F	G	H	I	
1	UDA/Area interviniente	Fecha de contacto in	Forma de contacto	Caracter del	Identidad se	Edad A	Vinculo institucional A	Pertenencia institucional /	Lugar de
2	Direccion de Genero y Diversidad	14/12/2021	Mail	Personal	Mujer cis	25	Nodocente	Misma que UDA/Area intervinie	Ciudad de
3	Direccion de Genero y Diversidad	11/10/2016	Encuentro presencial co	Personal	Mujer cis	23	Estudiante	Otra pertenencia	Ciudad de
4	Direccion de Genero y Diversidad	9/9/2017	Encuentro presencial co	Personal	Mujer cis	48	Nodocente	Otra pertenencia	Ciudad de
5	Direccion de Genero y Diversidad	1/6/2021	Derivacion institucional	Personal	Mujer cis	53	Estudiante	Otra pertenencia	Otra ciuda
6	Direccion de Genero y Diversidad	5/3/2020	Encuentro presencial co	Personal	Mujer cis	0	Se desconoce el dato	Otra pertenencia	
7	Direccion de Genero y Diversidad	22/8/2017	Encuentro presencial co	Personal	Mujer cis	20	Estudiante	Otra pertenencia	Otra ciuda
8	Direccion de Genero y Diversidad	14/3/2022	Encuentro presencial co	Personal	Mujer cis	34	Trabajador/a/e contratade	Misma que UDA/Area intervinie	Otra ciuda
9	Direccion de Genero y Diversidad	29/3/2022	Mail	Personal	Mujer cis	0	No tiene vinculo institucional	Otra pertenencia	Otra ciuda
10	Direccion de Genero y Diversidad	15/6/2016	Encuentro presencial co	Personal	Mujer cis	16	Estudiante	Otra pertenencia	Ciudad de
11	Direccion de Genero y Diversidad	10/4/2017	Derivacion institucional	Tercera persona	Mujer cis	19	Estudiante	Otra pertenencia	Ciudad de
12	Direccion de Genero y Diversidad	8/10/2018	Derivacion institucional	Personal	Mujer cis	0	Estudiante	Otra pertenencia	Ciudad de
13	Direccion de Genero y Diversidad	16/8/2019	Derivacion institucional	Personal	Mujer cis	24	Estudiante	Otra pertenencia	
14	Direccion de Genero y Diversidad	28/10/2020	Derivacion institucional	Personal	Mujer cis	0	Estudiante	Otra pertenencia	Otra ciuda
15	Direccion de Genero y Diversidad	23/03/2022	Encuentro presencial co	Personal	Varon trans		Docente	Otra pertenencia	Ciudad de
16	Direccion de Genero y Diversidad	13/4/2018	Encuentro presencial co	Personal	Mujer cis	29	Estudiante	Otra pertenencia	
17	Direccion de Genero y Diversidad	5/11/2018	Encuentro presencial co	Personal	Mujer cis	0	Estudiante	Otra pertenencia	Ciudad de
18	Direccion de Genero y Diversidad	2/3/2022	Derivacion institucional	Personal	Mujer cis	0	Nodocente	Otra pertenencia	Ciudad de
19	Direccion de Genero y Diversidad	3/10/2017	Encuentro presencial co	Personal	Mujer cis	33	Docente	Otra pertenencia	Ciudad de
20	Direccion de Genero y Diversidad	23/5/2019	Encuentro presencial co	Personal	Mujer cis	29	Graduade	Otra pertenencia	Otro país
21	Direccion de Genero y Diversidad	23/8/2017	Encuentro presencial co	Personal	Mujer cis	0	Estudiante	Otra pertenencia	
22	Direccion de Genero y Diversidad	21/2/2020	Mail	Personal	Mujer cis	40	Docente	Otra pertenencia	

Figura 5.2.1 - Hoja de cálculos de resultados

El formulario utilizado actualmente resulta limitado en cuanto a la visualización y análisis de datos, ya que solo proporciona una hoja de cálculo para el almacenamiento y procesamiento de la información recolectada junto con un conjunto muy acotado de visualizaciones.

La Dirección de Género, Diversidad y Derechos Humanos cuenta con la información recolectada por sus propios medios pero su procesamiento y visualización es una área de vacancia que le impide obtener información detallada sobre los patrones de violencia por razones de género en la Universidad, dificultando de este modo la toma de decisiones. PIBAS acerca una solución para visualizar y analizar la información.

5.3 Solución propuesta

Para abordar la problemática de forma más efectiva y completa, nace PIBAS como herramienta de análisis y visualización de datos. PIBAS permite tanto una gestión más eficiente de los datos como la generación de análisis y visualizaciones más profundas y significativas, lo que lleva a la toma de decisiones más informadas y efectivas en relación con la política institucional de género.

El formulario de Google existente se utilizó como contexto para la entrada de datos, configurando la herramienta de recolección de datos DEHIA con las preguntas definidas (Sección 5.3.1) y poblando con una muestra de casos reales relevados por las UDAS. Finalmente, la información importada es procesada por la plataforma PIBAS quedando disponibles potenciales visualizaciones con indicadores personalizables (Figura 4.2.2.4) por las personas de la Dirección y UDAS.

PIBAS permite una mayorflexibilidad en la forma en que se pueden visualizar los datos ya que, en lugar de depender de una hoja de cálculo estática, la solución propuesta permite generar visualizaciones interactivas que facilitan el análisis de los datos y la identificación de patrones y tendencias relevantes.

5.3.1 DEHIA

En esta sección vamos a mostrar brevemente la interacción necesaria con DEHIA para integrar al caso de estudio, de manera que se brinde el contexto suficiente para comprender su relación con la plataforma PIBAS.

DEHIA trabaja con los conceptos de ‘actividad’ y ‘tarea’, definiendo la instancia que engloba un objetivo y el conjunto de acciones a realizar para llevar a cabo el mismo, respectivamente. Bajo este esquema, se definió una actividad ‘Relevamiento caso violencia de género’ (Figura 5.3.1.1) y una tarea por pregunta existente en el formulario definido por las UDAS, las cualesfinalmente se asocian a la actividad creada inicialmente (Figura 5.3.1.2).

Figura 5.3.1.1 - Crear actividad en DEHIA

Figura 5.3.1.2 - Asociar tareas a una actividad

Las particularidades del formulario, en conjunto con las restricciones que impone DEHIA, requirieron tomar una serie de decisiones al momento de la configuración del mismo. Por un lado, en el formulario de las UDAS existen preguntas que admiten tanto respuestas cerradas (con opción predefinida) como respuestas abiertas (Figura 5.3.1.3), en su mayoría de carácter obligatorio. Mientras que, en la herramienta de recolección, la definición de las preguntas (mediante tareas) requiere establecer un tipo de tarea del conjunto*elegir una opción*, *elegir múltiple*, *simple (texto libre)*, entre otras, es decir restringe las respuestas a cerradas o abiertas. Ante estas condiciones, una pregunta como la de la Figura 5.3.1.3 no podría a priori representarse en DEHIA.

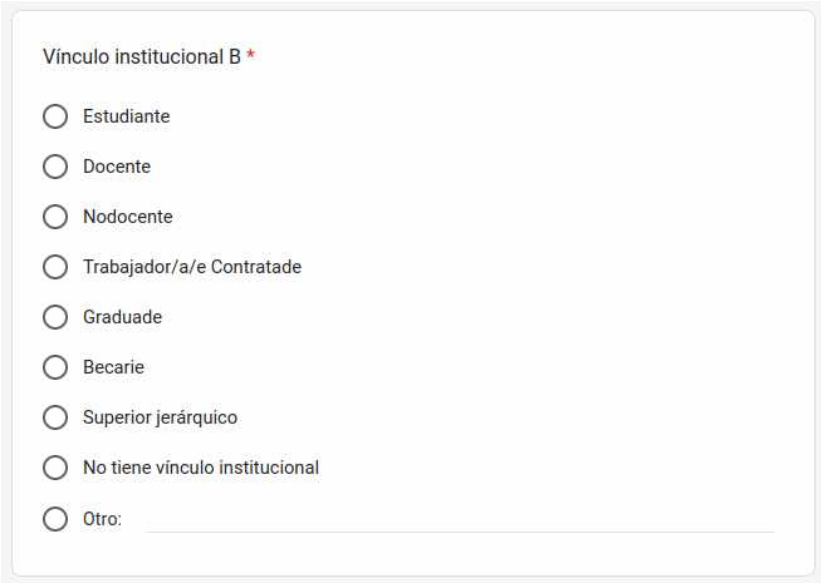


Figura 5.3.1.3 - Pregunta con respuesta cerrada o abierta

Para poder configurar el formulario de las UDAs respetando las necesidades de la herramienta de recolección de datos, se decidió reducir la tarea a un tipo de respuesta cerrada, generando adicionalmente una tarea de tipo*elegir opción*(*simple*o*múltiple* según lo defina la pregunta). Así, se establecen 2flujos posibles para la tarea generada adicionalmente según cómo se configure el tipo de planificación para la ejecución de las tareas asociadas que ofrece DEHIA. Ante un tipo de planificación 'secuencial' la tarea se ejecutará siempre y podrá requerir, como para el ejemplo (Figura 5.3.1.), definirla como opcional, mientras que con un tipo de planificación 'bifurcada' la tarea podrá ejecutarse dadas ciertas condiciones (previamente haber seleccionado una opción específica, etc.) y permitirfinalmente la duplicidad respuesta cerrada y/o abierta. Bajo

el objetivo de esta tesina, se decidió utilizar la tarea reducida a un tipo de respuesta cerrada, donde las respuestas abiertas quedan agrupadas por la categoría que definen las mismas (en el ejemplo, 'Otro').

Finalmente, el formulario de Google definido por las UDAS queda correctamente configurado y disponible en la herramienta DEHIA de manera que, una vez poblado con datos, pueda ser consumido por la plataforma PIBAS y procesado por las personas usuarias que originaron la instancia del mismo.

5.3.2 PIBAS

La aplicación que dispara el desarrollo de esta tesina es un prototipo construido con el objetivo de democratizar la creación de visualizaciones, a partir de los datos recolectados por las mismas personas usuarias, en este caso las UDAs. Para ello, se propuso una aplicación web que ofrece un conjunto de funcionalidades congregadas en 3 principales conceptos:*actividades*, *gráficos*y*tableros*(Figura 5.3.2.1).

Figura 5.3.2.1 - Menú

La sección actividades introduce una tabla que lista las instancias generadas y disponibles en DEHIA (Figura 4.2.2.3). En nuestro caso una instancia del formulario de las UDAs junto con las preguntas que lo conforman, visualizadas como 'tareas', y sus características (objetivo, dominio y tipo de planificación).

La sección gráficos es el apartado principal, el foco de la aplicación, en la cual incluso se apoyan las otras secciones núcleo de la herramienta. Aquí las personas usuarias podrán:

- Crear múltiples gráficos simultáneamente configurando individualmente el tipo de gráfico, las medidas con las que se debe construir y un nombre.
- Listar los gráficos generados por la persona usuaria.
- Ver un gráfico específico tanto en su valor histórico como en el actual. La diferencia entre el valor histórico y el actual de la visualización brinda consistencia en la información que se construye dado que los datos varían según las respuestas recolectadas, es decir en función del momento en que se procesen.
- Construir un nuevo gráfico en base al tipo y medidas de otro gráfico existente, permitiendo a las y los usuarios manejar cierta relación dentro de la construcción de distintas visualizaciones, en caso de requerir.
- Definir uno o más gráficos como favoritos para posicionarlos como un acceso rápido en la página de inicio.
- Descargar un gráfico, personalizando el título e incluso agregando una descripción para dar contexto al informe.

Dicha administración de gráficos se agrupa por actividad definida (Figura 5.3.2.2), entendiendo que las medidas bajo las cuales se regulará la construcción de visualizaciones debe darse bajo un mismo universo de datos.

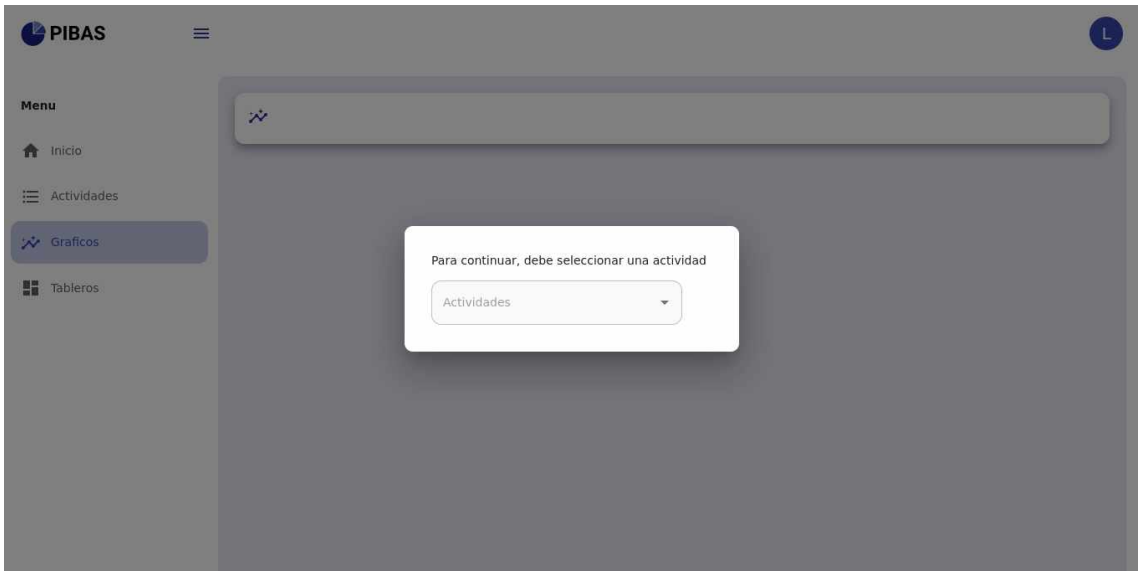


Figura 5.3.2.2 - Selección de actividad en sección gráficos

Por consiguiente, las medidas disponibles se constituyen de las tareas asociadas a la actividad (Figura 5.3.2.3), permitiendo seleccionar 1 misma tarea para las medidas columna yfila (ejexy ejey, respectivamente). Además, se establece una medida adicional de desglose para brindar a las personas usuarias un nivel más de agrupamiento e interseccionalidad entre los datos (Figura 5.3.2.4), cuyos valores continúan la lógica de la medida 'columna'.

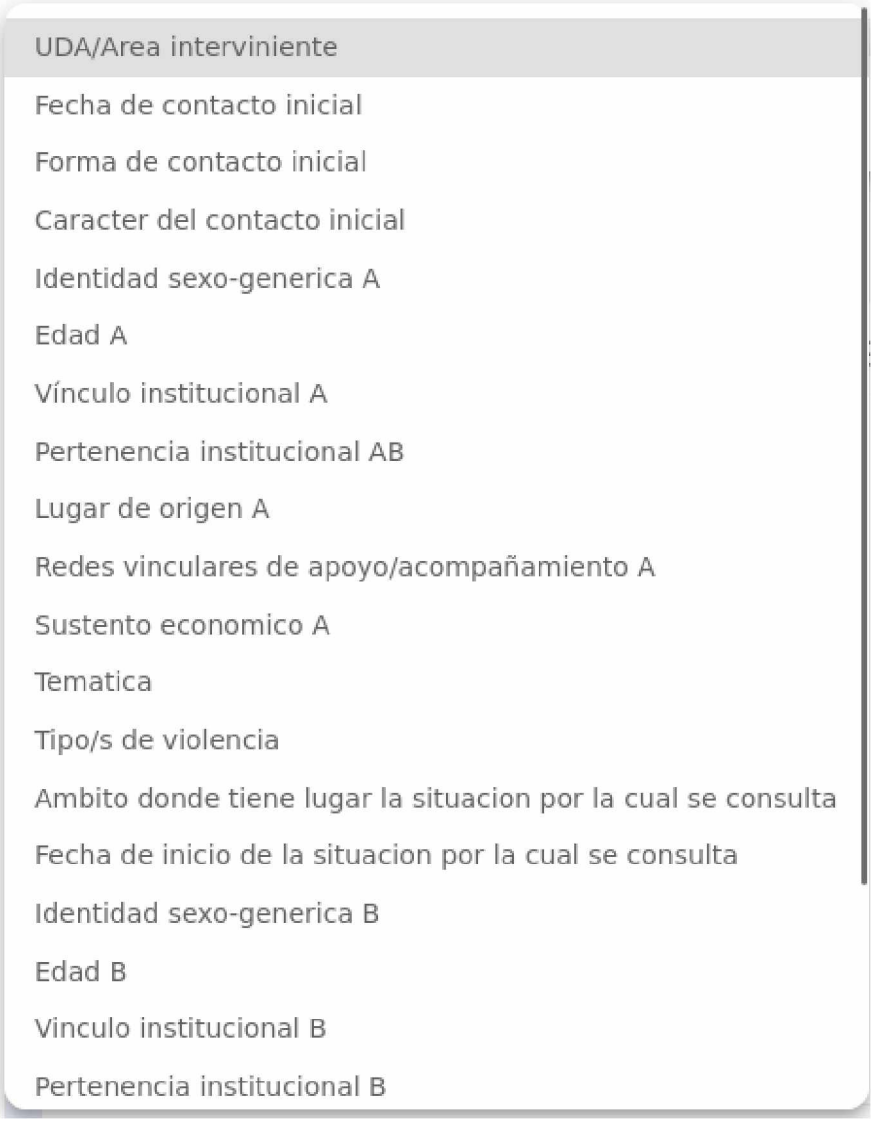


Figura 5.3.2.3 - Valores disponibles para las medidas

Figura 5.3.2.4 - Indicadores con medida de desglose

Respecto a los tipos de gráficos disponibles, se establecieron según un subconjuntofijo (Figura 5.3.2.5) de aquellos más utilizados en las herramientas de visualización analizadas en el Capítulo 3.

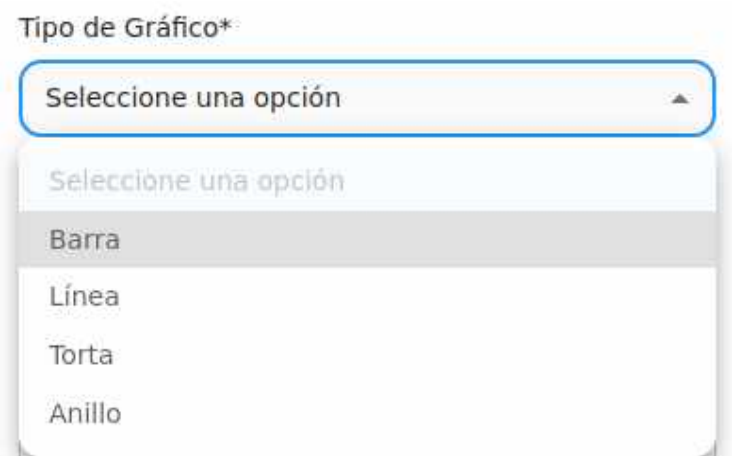


Figura 5.3.2.5 - Tipos de gráficos disponibles

Por último, como característica de la creación de gráficos, podemos mencionar el procesamiento de columnas yfilas. Para el eje x la plataforma ofrece configurar el dato con un subtipo específico, según el tipo de dato inicial en el que se establece la tarea seleccionada; por ejemplo, dado un dato de tipo "fecha" la persona usuaria puede configurar visualizar la información como "año" (Figura 5.3.2.6).

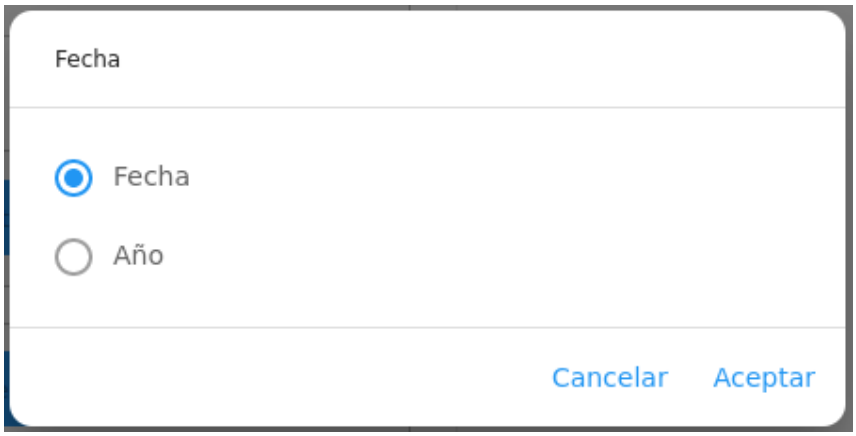


Figura 5.3.2.6 - Seleccionar el tipo de visualización para el eje x

En el caso del eje y las opciones de configuración no dependen del tipo de dato inicial, sino que está vinculado con el procesamiento de las tuplas para su visualización. De esta manera, todas las tareas configuradas en el eje y podrán visualizarse como: recuento, media o porcentaje (Figura 5.3.2.7).

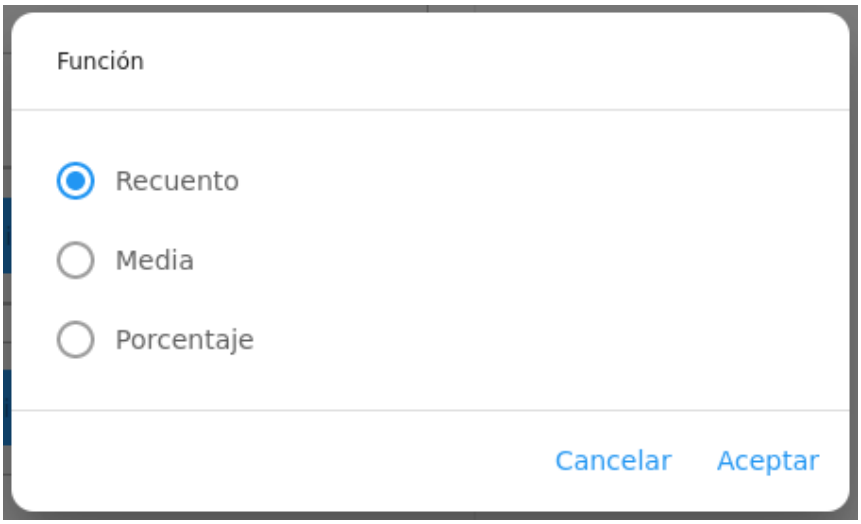


Figura 5.3.2.7 - Seleccionar el tipo de visualización para el eje y

En ambos casos, aunque en esta solución prototípica de PIBAS las opciones establecidas son acotadas, resulta interesante dejar planteado este nivel de configuración en el tipo de visualización de los ejes.

Por ejemplo, según lo mencionado hasta el momento, una persona usuaria de las UDAs podría generar un gráfico para visualizar información sobre las fechas de contacto inicial, de las situaciones registradas, según el porcentaje por año y agrupada por el vínculo institucional de la persona consultante (Figura 5.3.2.8).

Figura 5.3.2.8 - Porcentaje por año de contacto inicial y vínculo institucional de la persona consultante

La sección tableros permite crear conjuntos de visualizaciones y, opcionalmente, asignarles un título, así como editar cada instancia de tablero generada. Esta agrupación de gráficos no está regulada por actividad, por lo que implícitamente brinda una transversalidad de los 2 conceptos anteriores y, a su vez, construye una capa funcional más en la administración de visualizaciones. Además, para facilitar la selección de los gráficos que se quieren asignar al conjunto, la herramienta ofrece filtros por actividad, nombre de gráfico y favorito (Figura 5.3.2.9).

Figura 5.3.2.9 - Filtrar gráficos para la creación de un tablero

Cada tablero también incluye seleccionar como favorito, de la misma manera que para los gráficos, lo cual nos lleva a mencionar la sección*Inicio*visible en el menú de la Figura 5.3.2.1. Esta sección no engloba funcionalidades bajo un concepto sino que, por el contrario, da cierta información general del sistema (cantidad de actividades y de gráficos y tableros) y los recursos marcados como favoritos por la persona usuaria (Figura 4.2.2.1).

Por último, la plataforma PIBAS cierra el circuito de administración de visualizaciones con la posibilidad de descargar tanto un gráfico particular como un conjunto de gráficos agrupados en un tablero. Al momento de descargar, se direcciona a la persona usuaria a una previsualización del informe donde se le brinda la posibilidad de personalizar el título principal y, a su vez, agregar una descripción que pueda llegar a funcionar como contexto de la documentación a obtener (Figura 5.3.2.10).

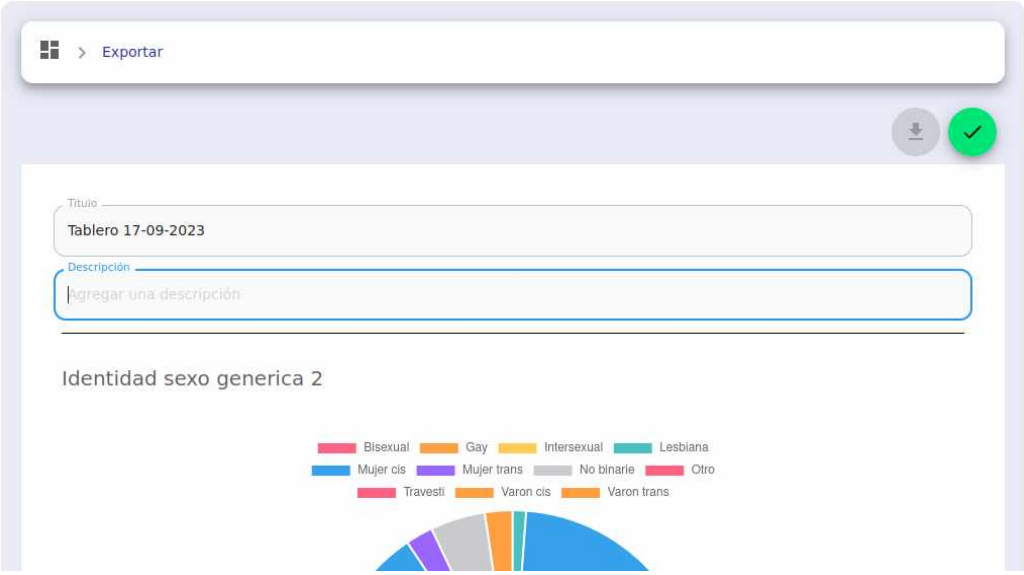


Figura 5.3.2.10 - Editar informe a descargar

Una vez que se ha personalizado el título principal y agregado una descripción contextual, el usuario puede proceder a generar y exportar el informe en formato PDF.

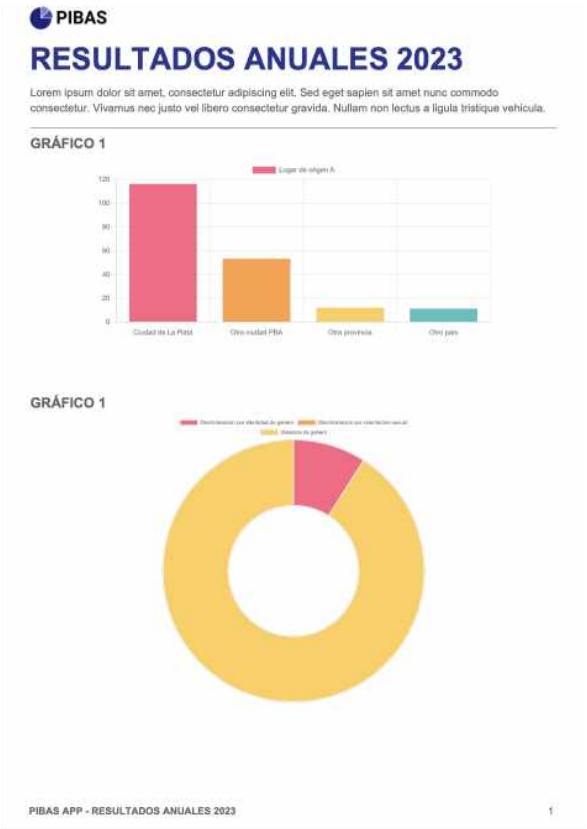


Figura 5.3.2.11 - Informe

Finalmente, la solución prototípica planteada ofrece la posibilidad de, unificadamente, recolectar datos a partir del formulario definido por la Dirección, generar gráficos para el análisis de los datos y la obtención de patrones (entre otra información potencial) de las situaciones que configuran violencia en el ámbito universitario, hacer un seguimiento de las visualizaciones en función de la recolección inicial y su evolución en el tiempo, por ejemplo a lo largo del ciclo lectivo, y personalizar conjuntos de visualizaciones según las necesidades de cada UDA y/o de la Dirección. Por último, dichos actores podrán concluir elflujo de administración de los datos y sus visualizaciones con la obtención de informes personalizados.

La solución propuesta ofrece un manejo completo,flexible y amplio en cuanto a la instanciación de visualizaciones en base a los datos recolectados. Al mismo tiempo, busca minimizar las complejidades en la interacción con la plataforma, como se mencionó en la sección 4.2.2 en relación al diseño de interfaz de usuario, sopesando aquellas complejidades intrínsecas en la construcción de visualizaciones.

5.4 Recapitulación

En el capítulo se presentó el caso de estudio. En el último capítulo, que sigue a continuación, se presentan las conclusiones a las que se llegó con el desarrollo de esta plataforma, así como aspectos que quedaron pendientes y que quedan planteados como trabajos futuros.

Capítulo 6: Conclusiones y trabajos futuros

En este capítulo se presentan las conclusiones obtenidas a partir de la investigación realizada en el campo de la visualización de datos, así como las posibles líneas de trabajo futuro que se podrían abordar.

6.1 Conclusiones

En primer lugar, llevamos a cabo un estudio detallado del estado del arte en relación con plataformas de visualización y análisis de datos lo que nos permitió identificar las herramientas y tecnologías más usadas en el mercado y las tendencias en cuanto a diseño y funcionalidades.

En segundo lugar, definimos la arquitectura de la plataforma PIBAS y se integró con DEHIA, permitiendo la conexión y la manipulación de datos entre ambas plataformas. Además, se definieron los requisitos funcionales y no funcionales de la plataforma, con elfin de garantizar la eficacia y eficiencia en su implementación.

En tercer lugar, se desarrolló una solución prototípica que implementa la plataforma PIBAS, la cual incluye un conjunto de visualizaciones interactivas y herramientas de análisis de datos. Esta solución prototípica permitió validar la arquitectura propuesta y las funcionalidades implementadas.

Por último, aplicamos el prototipo al dominio del Programa Institucional Contra la Violencia de Género en el ámbito de la Universidad Nacional de La Plata. Esto nos permitió poner a prueba la solución prototípica y obtener resultados concretos en cuanto a la eficacia y eficiencia de la plataforma en un caso de uso real.

En conclusión, se ha logrado el objetivo propuesto en la tesina, que consistió en diseñar e implementar una plataforma de visualización y análisis de datos para el Programa Institucional Contra la Violencia de Género en el ámbito de la Universidad Nacional de La Plata. La plataforma PIBAS propuesta permite visualizar y analizar los datos de

manera interactiva y eficiente, y puede ser utilizada para tomar decisiones informadas en la lucha contra la violencia de género.

6.2 Contribuciones

Durante el transcurso del desarrollo de la tesina se presentó, y se expuso a evaluación un artículo relacionado al tema de la tesina de grado. El cual ha sido aceptado y luego expuesto por quienes presentan esta tesina:

Acha Altamiranda Pilar, Rodriguez Pau Lola, Arcidiacono Jose, Bazán Patricia y Lliteras Alejandra B.**PIBAS: visualización de datos con perspectiva de género**en JAIIO 2022, Simposio Argentino sobre Tecnología y Sociedad.

Abstract.

La plataformización de las actividades de la sociedad hace que cada vez se recolecten más y más datos, los formularios creados ad-hoc para analizar problemáticas sociales, como por ejemplo, la violencia de género, ponen de relieve la necesidad de dar soporte a la sociedad toda, para que pueda visualizar y analizar los datos recolectados, fundamentalmente para apoyar la toma de decisiones y la generación de nuevas políticas públicas (o la aplicación de las existentes) en los contextos más vulnerables.

Considerando el principio de simetría de la Teoría Actor-Red (TAR), los datos son actores fundamentales a considerar en todo estudio, en particular, relacionado a las humanidades digitales y la democratización de su visualización para personas usuarias sin conocimientos previos de informática, un derecho.

En este trabajo se presenta una plataforma de visualización de datos para perosnas usuarias finales que fue utilizada para analizar un caso de estudio a partir de datos obtenidos por las Unidades de Atención del Programa y del Protocolo de actuación ante situaciones de discriminación y/o violencia de género de la Universidad Nacional de La Plata (UDAS).

En este artículo se presenta un breve análisis de diversas soluciones y plataformas que permiten a las personas usuariasfinales visualizar información sin necesidad de tener conocimientos específicos de visualización de datos (Capítulo 3). Describe en detalle la plataforma de visualización desarrollada, explica la arquitectura distribuida de PIBAS, su interfaz de usuario, las funcionalidades que ofrece y cómo se presenta la

visualización de datos a través de gráficos (Capítulo 4). Se utiliza el caso de estudio de la Dirección de Género y Diversidad de la UNLP para demostrar el potencial de la plataforma en la visualización y análisis de datos (Capítulo 5). Describe conclusiones sobre la integración de la plataforma con DEHIA, la aplicación en el caso de estudio y la importancia de la visualización de datos en la toma de decisiones y políticas públicas (Capítulo 6).

6.3 Trabajos futuros

En este capítulo, se presentan diversas direcciones y áreas de mejora que podrían ser consideradas para futuros desarrollos y expansiones de la aplicación de generación de gráficos. Estas ideas representan oportunidades para aumentar la funcionalidad, la inteligencia y la seguridad de la aplicación, así como para proporcionar una experiencia de usuario más completa y avanzada.

6.3.1 Recomendaciones de Gráficos Inteligentes

Uno de los enfoques para mejorar la aplicación es la implementación de un sistema de recomendación de gráficos inteligentes. Este sistema utilizaría algoritmos de inteligencia artificial para analizar los datos proporcionados por el usuario y, en función de la naturaleza de los datos y el contexto del informe, sugeriría tipos de gráficos apropiados. La capacidad de la aplicación para comprender la mejor representación gráfica de los datos proporcionaría una experiencia más eficiente y efectiva para los usuarios.

6.3.2 Personalización Avanzada de Gráficos

Ampliar las capacidades de personalización de gráficos, permitiendo a los usuarios definir reglas de personalización más complejas y personalizadas. Por ejemplo, permitir la personalización de colores, etiquetas y estilos de acuerdo con variables específicas.

6.3.3 Colaboración en Tiempo Real

Agregar características de colaboración en tiempo real para permitir que varios usuarios trabajen juntos en un informe o tablero, compartiendo y editando gráficos de forma simultánea.

6.3.4 Historial de Versiones de Gráficos

Actualmente, la aplicación permite a los usuarios visualizar el gráfico que se creó y el gráfico correspondiente al día de hoy. Sin embargo, para mejorar la experiencia y brindar una funcionalidad más completa, se puede considerar la implementación de un historial de versiones de gráficos. Esta función permitiría a los usuarios realizar un seguimiento y acceder a versiones anteriores de los gráficos que han creado a lo largo del tiempo.

6.3.5 Control de Acceso Basado en Roles

Implementar un sistema de control de acceso basado en roles que permita asignar permisos específicos a usuarios y grupos para garantizar que solo las personas autorizadas tengan acceso a determinadas funciones y datos.

Referencias bibliográficas

[Arcidiacono, 2020] Aricidiácono Jose (2020). DEHIA: Una plataforma liviana para definir y ejecutar actividades con intervención humana basadas en workflows. http://sedici.unlp.edu.ar/handle/10915/116617

[Few, 2013] Few, Stephen (2013). Information Dashboard Design: Displaying Data for At-a-Glance Monitoring.

[Healy, 2018] Healy, Kieran. (2018). Data Visualization: A Practical Introduction. Princeton University Press.

[Khalid & Zebaree, 2021] Khalid, Z. M., & Zebaree, S. R. (2021). Big data analysis for data visualization: A review. International Journal of Science and Business, 5(2), 64-75.

[Knaflic, 2023] Knaflic, C. N. (2022). Storytelling with you: Plan, Create, and Deliver a Stellar Presentation. John Wiley & Sons.

[Mayer-Schönberger and Cukier, 2013] Mayer-Schönberger, V., & Cukier, K. (2013). Big data: la revolución de los datos masivos. Turner.

[Sosulski, 2018] Sosulski, Kristen. (2019). Data visualization made simple: Insights Into Becoming Visual.

[Tanenbaum & Van Steen, 2017] Tanenbaum, A. S., & Van Steen, M. (2017). Distributed Systems: Principles and Paradigms. Pearson.

[Tufte E. R. , 2001] Tufte, E. R. (2001). The visual display of quantitative information. Graphics Press.

[Unwin, 2020] Unwin, A. (2020). Título del libro. Ciudad: Editorial.

[Van Dijck et all., 2018] Van Dijck, J., Poell, T., & De Waal, M. (2018). The platform society: Public values in a connective world. Oxford University Press.

[Wilkinson, 2013] Wilkinson, Leland (2013). The Grammar of Graphics. Springer.